REPORT DOCUMENTATION PAGE	Form Approved OMB NO. 0704-0188

The public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggesstions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington VA, 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to any oenalty for failing to comply with a collection of information if it does not display a currently valid OMB control number.
PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS.

1. REPORT DATE (DD-MM-YYYY) 11-03-2016	2. REPORT TYPE Ph.D. Dissertation	3. DATES COVERED (From - To) -

4. TITLE AND SUBTITLE SPEECH DATA ANALYSIS FOR SEMANTIC INDEXING OFVIDEO OF SIMULATED MEDICAL CRISES	5a. CONTRACT NUMBER W911NF-13-1-0066
	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHORS Shuangshuang Jiang	5d. PROJECT NUMBER
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAMES AND ADDRESSES University of Louisville 2301 S. Third Street Jouett Hall Louisville, KY 40208 -1838	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS (ES) U.S. Army Research Office P.O. Box 12211 Research Triangle Park, NC 27709-2211	10. SPONSOR/MONITOR'S ACRONYM(S) ARO 11. SPONSOR/MONITOR'S REPORT NUMBER(S) 63184-CS.16

12. DISTRIBUTION AVAILIBILITY STATEMENT

Approved for public release; distribution is unlimited.

13. SUPPLEMENTARY NOTES

The views, opinions and/or findings contained in this report are those of the author(s) and should not contrued as an official Department of the Army position, policy or decision, unless so designated by other documentation.

14. ABSTRACT

This dissertation introduces our developed system for efficient segmentation and semantic indexing of videos of medical simulations using machine learning methods. It provides the physician with automated tools to review important sections of the simulation by identifying who spoke, when and what was his/her emotion. Only audio information is extracted and analyzed because the quality of the image recording is low and the visual environment is static for most parts. Our proposed

15. SUBJECT TERMS

Speech segmentation; speaker recognition; classification; feature extraction

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	15. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			Hichem Frigui
UU	UU	UU	UU		19b. TELEPHONE NUMBER 502-852-2009

Standard Form 298 (Rev 8/98)
Prescribed by ANSI Std. Z39.18

Report Title

SPEECH DATA ANALYSIS FOR SEMANTIC INDEXING OFVIDEO OF SIMULATED MEDICAL CRISES

ABSTRACT

This dissertation introduces our developed system for efficient segmentation and semantic indexing of videos of medical simulations using machine learning methods. It provides the physician with automated tools to review important sections of the simulation by identifying who spoke, when and what was his/her emotion. Only audio information is extracted and analyzed because the quality of the image recording is low and the visual environment is static for most parts. Our proposed system includes four main components: preprocessing, speaker segmentation, speaker identification, and emotion recognition. The preprocessing consists of first extracting the audio component from the video recording. Then, extracting various low-level audio features to detect and remove silence segments. We investigate and compare two different approaches for this task. The first one is threshold-based and the second one is classification-based. The second main component of the proposed system consists of detecting speaker changing points for the purpose of segmenting the audio stream. We propose two fusion methods for this task.

The speaker identification and emotion recognition components of our system are designed to provide users the capability to browse the video and retrieve shots that identify "who spoke, when, and the speaker's emotion" for further analysis. For this component, we propose two feature representation methods that map audio segments of arbitary length to a feature vector with fixed dimensions. The first one is based on soft bag-of-word (BoW) feature representations. In particular, we define three types of BoW that are based on crisp, fuzzy, and possibilistic voting. The second feature representation is a generalization of the BoW and is based on Fisher Vector (FV). FV uses the Fisher Kernel principle and combines the benefits of generative and discriminative approaches. The proposed feature representations are used within two learning frameworks. The first one is supervised learning and assumes that a large collection of labeled training data is available. Within this framework, we use standard classifiers including K-nearest neighbor (K-NN), support vector machine (SVM), and Naive Bayes. The second framework is based on semi-supervised learning where only a limited amount of labeled training samples are available. We use an approach that is based on label propagation.

SPEECH DATA ANALYSIS FOR SEMANTIC INDEXING OF VIDEO OF SIMULATED MEDICAL CRISES

By

Shuangshuang Jiang
B.S., EE, Huazhong University of Science and Technology, 2007
M.S., Information System, Wuhan University, 2009

A Dissertation
Submitted to the Faculty of the
J.B. Speed School of Engineering of the University of Louisville
in Partial Fulfillment of the Requirements
for the Degree of

Doctor of Philosophy in Computer Science and Engineering

Department of Computer Engineering and Computer Science
University of Louisville
Louisville, Kentucky

May 2015

Copyright 2015 by Shuangshuang Jiang

All rights reserved

SPEECH DATA ANALYSIS FOR SEMANTIC INDEXING OF VIDEO OF SIMULATED MEDICAL CRISES

By

Shuangshuang Jiang
B.S., EE, Huazhong University of Science and Technology, 2007
M.S., Information System, Wuhan University, 2009

A Dissertation Approved On

April 23, 2015

by the following Dissertation Committee:

Hichem Frigui, Ph.D., Dissertation Director

Aaron W. Calhoun, MD

Tim Hardin, Ph.D.

Adrian Lauf, Ph.D.

Olfa Nasraoui, Ph.D.

Roman V. Yampolskiy, Ph.D.

ACKNOWLEDGEMENTS

I would like to express my sincere appreciation and respect to my advisor Dr. Hichem Frigui for his support, guidance and patience during all the time that I have spent working on this thesis. He provided me an excellent atmosphere for doing research and developing ideas, that became the building blocks of my dissertation. Without his help and expertise it would simply be impossible to accomplish this work, and I feel lucky that I had this opportunity to work in his team.

I would like to thank Dr. Aaron W. Calhoun for giving me the opportunity to work on this project, providing data and valuable suggestions, and for serving on my dissertation committee.

I would also like to thank all other members of my dissertation committee Dr. Tim Hardin, Dr. Adrian Lauf, Dr. Olfa Nasraoui, and Dr. Roman V. Yampolskiy for their time, service, comments, and valuable suggestions.

Great thanks goes to all the members of the Multimedia Research Laboratory who have worked shoulder to shoulder with me during these years: Hamdi Jenzri, Oualid Missaoui, Andrew D Karem, Lijun Zhang, Jason Meredith, Anis Hamdi, Naouel Baili, Mohamed Ben Ismail, Ahmed Chamseddine, Ouiem Bchir, Surangkana Rawungyot, Faezeh Tafazzoli, and Amine Ben Khalifa.

I would like to acknowledge the help of my dearest friends Dongqing Chen, Hui Wang, Yinan Cui, Jian Wang, Wei Zhou, Liang Zhang, Li Zhu, Jubin Chen, Wenlong Sun, Hanwen Yuan, Xue Shi, and Xiang Zhang for their friendship and support. They kept my motivation high during the times when I was not on my top.

I would like to dedicate this thesis to my family. The love of my parents, wife, and daughter have always been those essential things that bring sense in my life.

ABSTRACT

SPEECH DATA ANALYSIS FOR SEMANTIC INDEXING OF VIDEO OF SIMULATED MEDICAL CRISES

Shuangshuang Jiang

April 23, 2015

The Simulation for Pediatric Assessment, Resuscitation, and Communication (SPARC) group within the Department of Pediatrics at the University of Louisville, was established to enhance the care of children by using simulation based educational methodologies to improve patient safety and strengthen clinician-patient interactions. After each simulation session, the physician must manually review and annotate the recordings and then debrief the trainees. The physician responsible for the simulation has recorded 100s of videos, and is seeking solutions that can automate the process.

This dissertation introduces our developed system for efficient segmentation and semantic indexing of videos of medical simulations using machine learning methods. It provides the physician with automated tools to review important sections of the simulation by identifying who spoke, when and what was his/her emotion. Only audio information is extracted and analyzed because the quality of the image recording is low and the visual environment is static for most parts. Our proposed system includes four main components: preprocessing, speaker segmentation, speaker identification, and emotion recognition. The preprocessing consists of first extracting the audio component from the video recording. Then, extracting various low-level audio features to detect and remove silence segments. We investigate and compare two different approaches for this task. The first one is threshold-based and the second one is classification-based. The second main component of the proposed system consists of detecting speaker changing points for the purpose of segmenting the audio stream. We propose two fusion methods for this task.

The speaker identification and emotion recognition components of our system are designed to provide users the capability to browse the video and retrieve shots that identify ”who spoke, when, and the speaker’s emotion” for further analysis. For this component, we propose two feature representation methods that map audio segments of arbitary length to a feature vector with fixed dimensions. The first one is based on soft bag-of-word (BoW) feature representations. In particular, we define three types of BoW that are based on crisp, fuzzy, and possibilistic voting. The second feature representation is a generalization of the BoW and is based on Fisher Vector (FV). FV uses the Fisher Kernel principle and combines the benefits of generative and discriminative approaches. The proposed feature representations are used within two learning frameworks. The first one is supervised learning and assumes that a large collection of labeled training data is available. Within this framework, we use standard classifiers including K-nearest neighbor (K-NN), support vector machine (SVM), and Naive Bayes. The second framework is based on semi-supervised learning where only a limited amount of labeled training samples are available. We use an approach that is based on label propagation.

Our proposed algorithms were evaluated using 15 medical simulation sessions.

The results were analyzed and compared to those obtained using state-of-the-art algorithms. We show that our proposed speech segmentation fusion algorithms and feature mappings outperform existing methods. We also integrated all proposed algorithms and developed a GUI prototype system for subjective evaluation. This prototype processes medical simulation video and provides the user with a visual summary of the different speech segments. It also allows the user to browse videos and retrieve scenes that provide answers to semantic queries such as: who spoke and when; who interrupted who? and what was the emotion of the speaker? The GUI prototype can also provide summary statistics of each simulation video. Examples include: for how long did each person spoke? What is the longest uninterrupted speech segment? Is there an unusual large number of pauses within the speech segment of a given speaker?

TABLE OF CONTENTS

ACKNOWLEDGEMENTS iii
ABSTRACT v
LIST OF TABLES xi
LIST OF FIGURES xii
LIST OF ALGORITHMS xvii

CHAPTER

1 INTRODUCTION 1
  1.1 Motivations 1
  1.2 Contributions 3

2 RELATED WORK 6
  2.1 Overview of A Typical Speech Data Analysis System 6
  2.2 Feature Representation 8
    2.2.1 Feature Extraction 8
    2.2.2 Feature Preprocessing 15
  2.3 Speaker Segmentation 15
    2.3.1 Segmentation Algorithms 16
    2.3.2 Segment Feature Representation 27
    2.3.3 Distance Measures 30
  2.4 Speaker Recognition 30
  2.5 Bag-of-words Feature Representation 32
  2.6 Emotion Recognition 34
  2.7 Spoken Document Retrieval (SDR) 36

3 SPEAKER SEGMENTATION 38

3.1 Motivations 38

3.2 Extrema Point-level Fusion 39

3.3 Distance-level Fusion 40

4 SPEAKER IDENTIFICATION 44

4.1 Motivations 44

4.2 Soft Bag-of-words Feature Representation 46

4.2.1 Visual Vocabulary Construction 46

4.2.2 Membership Mapping and Feature Representation 47

4.2.3 Crisp Mapping 48

4.2.4 Fuzzy Mapping 48

4.2.5 Possibilistic Mapping 49

4.3 Fisher Vector Feature Representation 50

4.3.1 Fisher Kernel and Fisher Vector 51

4.3.2 Fisher Vector Features for Speech Data 52

4.4 Classification Algorithms for Speaker Identification 54

4.4.1 K-NN classifier 54

4.4.2 Naive Bayes Classifier 55

4.4.3 Support Vector Machines (SVM) Classifier 56

5 SEMI-SUPERVISED SPEAKER IDENTIFICATION 57

5.1 Motivations 57

5.2 Label Propagation 58

5.2.1 Label Propagation Algorithm 58

5.3 Speaker Identification with Label Propagation 60

6 EXPERIMENTAL RESULTS AND ANALYSIS 62
6.1 Data Collections 62
6.2 Data Preprocessing 62
6.3 Detection and Removal of Silence Segments 64
6.3.1 Threshold-based Silence Detection 64
6.3.2 Classification-based Silence Detection 66
6.4 Feature Extraction and Mapping 70
6.4.1 Low-level Features 70
6.4.2 Soft Bag-of-Words Feature Mapping 71
6.4.3 Fisher Vector Feature Mapping 72
6.5 Speaker Segmentation 75
6.6 Speaker Identification 76
6.7 Emotion Recognition 77
6.8 Results and Analysis 77
6.8.1 Speaker Segmentation Algorithms Used for Comparison . 77
6.8.2 Evaluation Measures 80
6.8.3 Results and Discussion 82
6.9 Content-based Segmentation and Retrieval of Medical Simulation Video 105
7 CONCLUSIONS AND POTENTIAL FUTURE WORK 111
7.1 Conclusions 111
7.2 Potential Future Work 113
REFERENCES 115
CURRICULUM VITAE 125

LIST OF TABLES

TABLE Page
6.1 Data collections used to analyze and evaluate the various speaker segmentation/recognition algorithms. 63
6.2 Speaker segmentation results by 7 different algorithms. For each algorithm, we report the number of detected changing points and the number of true changes in (). 91
6.3 Classification rate of the K-NN classifier using the proposed soft bag of words representation of MFCC features and various distance measures 95
6.4 EMO-DB description 102
6.5 EMO-DB speakers emotion recognition based on MFCC feature . . . 103
6.6 EMO-DB speakers emotion recognition based on PLP feature 104
6.7 EMO-DB speakers emotion recognition based on LPCC feature . . . 104
6.8 EMO-DB speakers emotion recognition based on GFCC feature . . . 104

LIST OF FIGURES

FIGURE Page

2.1 Overview of a typical speech data analysis system. 6

2.2 Different steps involved in extracting the MFCC features. 9

2.3 A triangular Mel-filter bank. 10

2.4 Different steps involved in the extraction of RASTA-PLP features. . . 13

2.5 Architecture of a typical speaker segmentation system. 16

2.6 BIC curves of two uterances with: (a) no changing point, (b) one changing point. 20

2.7 Diagram of $WinGrow$ for speaker change detection. 21

2.8 Recursive tree that simulates the recursive process of the DAC algorithms. 25

3.1 Speaker changing points detected by BIC, KL, and T^2 algorithms. Circled points are the true change points. 40

3.2 Distance curves for speaker changing points detection by BIC, KL, T^2 and the proposed distance level fusion. 42

4.1 An overview of the speaker identification component of the proposed system. 45

6.1 Threshold-based Silence Detection. (a) original STE values (in blue) and filtered STE values (in red) by a moving average smoothing; (b) original SC values (in blue) and filtered SC values (in red) by a moving average smoothing; (c) silence detection results, red represents speech sections and gray denotes detected silence. 65

6.2 An example where threshold-based silence detection methods perform poorly. 67

6.3 A scatter plot of the three features used to discriminate between silence and speech audio segments. 68

6.4 Silence detection results based on SVM classification for the audio stream used in Fig. 6.1. 68

6.5 Silence segments detected using (a) threshold-based method and (b) classification-based method for noisy audio stream 1. 69

6.6 Silence segments detected using (a) threshold-based method and (b) classification-based method for noisy audio stream 2. 70

6.7 Main steps involved in extracting low-level audio features. 71

6.8 Response of all prototypes (before merging) to 4 segments from different speakers. (a) speaker 1, (b) speaker 2, (c) speaker 3, (d) speaker 4. 73

6.9 Response of all prototypes (after merging) to 4 segments from different speakers. (a) speaker 1, (b) speaker 2, (c) speaker 3, (d) speaker 4. . 74

6.10 Evaluation measures of *ChenBIC* speaker segmentation method on *Med*2 audio data with 12dim MFCC features. 83

6.11 Evaluation measures of four speaker segmentation methods on *Med*2 audio data with 12dim MFCC features. (a) *SeqBIC* method, (b) *DAC*1 method, (c) *DAC*2 method, (d) *DAC*3 method. 84

6.12 Detected changing points by *ChenBIC* and real changing points in *Med*2 audio data. (a) results on the entire audio signal, (b) details of the results from 215s to 265s. 85

6.13 Detected changing points by $DAC3$ and real changing points in $Med2$ audio data. (a) results on the whole audio signal, (b) details of the results of the audio segment from 215s to 265s. 86
6.14 Evaluations of DAC3 speaker segmentation method on $Med2$ audio data using MFCC features with (a) 12 dimension, (b) 24 dimension, and (c) 36 dimension. 87
6.15 Evaluations of DAC3 speaker segmentation method on $Med2$ audio data using PLP features with (a) 12 dimension, (b) 20 dimension, and (c) 23 dimension. 87
6.16 Evaluations of DAC3 based speaker segmentation methods on $Med2$ audio data with (a) MFCC(12dim)+Δ, and (b) MFCC(12dim)+Δ+$\Delta\Delta$ features. 88
6.17 Evaluations of DAC3 based speaker segmentation methods on $Med2$ audio data with (a) PLP(12dim)+Δ, and (b) PLP(12dim)+Δ+$\Delta\Delta$ features. 88
6.18 Comparison of the FAR (first bar) and RCL (second bar) of seven speaker segmentation methods on $Med2$ audio data with 12dim MFCC features when $\lambda = 1$. 89
6.19 Statistics of the RCL measure values over the 15 data sets of the seven methods with MFCC and PLP features. For each algorithm, the box represents the statistics of the values, the red line is the median, the edges of the box are the 25th and 75th percentiles. 92
6.20 Statistics of the FAR measure values over the 15 data sets of the seven methods with MFCC and PLP features. 93
6.21 Performance of the crisp, fuzzy, and possibilistic BoW using MFCC features with the KNN, SVM, and NB classifiers 95

6.22 Performance of the Fisher Vector representation using MFCC, PLP, and LPCC features with the KNN and SVM classifiers 96

6.23 Comparison of the classification accuracy of soft BoW feature mappings (C-BoW, F-BoW, and P-BoW) using the NB classifier and FV feature mapping with KNN classifier with GMM-UBM, GMM mean supervector with K-NN (SV-KNN) and SVM (SV-SVM) using MFCC features. The results are averaged over 15 datasets. 97

6.24 Comparison of the classification accuracy of soft BoW feature mappings (C-BoW, F-BoW, and P-BoW) using the NB classifier and FV feature mapping with KNN classifier with GMM-UBM, GMM mean supervector with K-NN (SV-KNN) and SVM (SV-SVM) using PLP features. The results are averaged over 15 datasets. 97

6.25 Comparison of the classification accuracy of soft BoW feature mappings (C-BoW, F-BoW, and P-BoW) using the NB classifier and FV feature mapping with KNN classifier with GMM-UBM, GMM mean supervector with K-NN (SV-KNN) and SVM (SV-SVM) using LPCC features. The results are averaged over 15 datasets. 98

6.26 Comparison of the classification accuracy of soft BoW feature mappings (C-BoW, F-BoW, and P-BoW) using the NB classifier and FV feature mapping with KNN classifier with GMM-UBM, GMM mean supervector with K-NN (SV-KNN) and SVM (SV-SVM) using GFCC features. The results are averaged over 15 datasets. 98

6.27 Classification accuracy of the FV feature mapping with a K-NN classifier using MFCC features, as we vary the percentage of labeled data. 100

6.28 Classification accuracy of the FV feature mapping with a K-NN classifier using PLP features, as we vary the percentage of labeled data. . 101

6.29 Classification accuracy of the FV feature mapping with a K-NN classifier using LPCC features, as we vary the percentage of labeled data. 101
6.30 Comparison the speaker identification accuracy using standard supervised learning with a K-NN classifier with semi-supervised learning with 80% labeled samples. 102
6.31 Flowchart of developed GUI to aid the physician in reviewing medical simulation data. 106
6.32 Initial interface of the GUI with descriptor of its 7 panels. 107
6.33 Various parameters that the user can modify. Each parameter has a default value that was optimized in our experiment. 108
6.34 Visualization of the speaker identification results. 109
6.35 Visualization of the emotion recognition results. 110

LIST OF ALGORITHMS

ALGORITHM Page

2.1 Speaker Segmentation by $ChenBIC$ algorithm 20

2.2 Speaker Segmentation by $SeqBIC$ algorithm 22

2.3 Speaker Segmentation by $DAC1$ algorithm 23

2.4 Speaker Segmentation by $DAC2$ algorithm 23

2.5 Speaker Segmentation by $DAC3$ algorithm 24

3.1 Extrema point-level fusion 41

3.2 Distance level fusion 43

5.1 Fisher Vector-based Speaker Identification using Semi-supervised Learning with Label Propagation 61

CHAPTER 1

INTRODUCTION

1.1 Motivations

Many studies have confirmed the $volume - outcome$ principle, which states that centers with higher volumes of a given condition typically have better outcomes for that disorder. This phenomenon is of crucial importance in pediatric education, as medical crises are rare events and thus, can generate potentially crippling anxiety when encountered. Medical simulations, where uncommon clinical situations can be replicated for educational purposes, have proved to provide more consistent training of clinicians. Consequently, the Simulation for Pediatric Assessment, Resuscitation, and Communication (SPARC) group, within the Department of Pediatrics at the University of Louisville, was established. The SPARC group is composed of a multidisciplinary group of physicians, nurses, and respiratory therapists. It exists to enhance the care of infants and children by using simulation-based educational methodologies to improve patient safety, strengthen interdisciplinary and clinician-patient interactions, engage in local and reginal outreach, and disseminate innovative curricula. These sessions involve 4 to 9 people and last 20 minutes to one hour. They are scheduled approximately twice per week and are recorded as video data.

During each session, the physician/instructor must manually review and annotate the recording in real time and then debrief the trainees on the session immediately following its resolution. This video debriefing is considered a crucial part of the educational process as it allows participants to actively reflect on their performance and potentiates behavioral changes. To date, however, this video data has gone largely unused due to the labor-intensive nature of the manual review and segmentation processes, as well as the difficulty in quickly identifying and moving to key images or events. This effectively prevents the SPARC program from using one of their most valuable educational tools most effectively.

Providing the physician with automated tools to segment, semantically index and retrieve specific scenes from a large database of training sessions offers an innovative solution to this issue by enabling him/her to immediately review important sections of the training with the team. Thus, allowing the dissemination of more efficient debriefing sessions with the team of trainees. A further benefit is the potential to enable the rapid identification of similar circumstances in previously recorded sessions (the SPARC program currently has over 90 sessions recorded in DVD format). This would potentiate the discovery of critical similarities that are common across training sessions that could then be used to predict outcome.

Such an innovation would also have repercussions far beyond the SPARC program. As simulation becomes more entrenched into medical education, the need for software that can automate crucial aspects of the process and free up instructor time for more educationally useful activities will only rise. Currently, many smaller institutions are attempting to launch simulation programs with only minimal staff and space. The SPARC program is an example of one such endeavor. Developing software to enable the rapid segmentation of video data would enable offering better education using fewer faculty instructors than have been needed in the past. In addition, many hospitals that would benefit from simulation educational outreach programs but are unable to provide their own simulation programs, would also benefit from such an automated system.

At present, however, no software exists to enhance this process, and hence all analysis is done by hand. The analysis tools we propose to develop would fill this gap, allowing for greater analytical efficiency, potentially allowing information to be delivered to code participants before their shift ends, when they would most significantly benefit from it. In particular, we propose methods to automate the analysis speech data in medical video simulations. These methods include speaker segmentation, speaker diarization, speaker recognition, and emotion recognition.

1.2 Contributions

This dissertation addresses the development of effective tools for the extraction, integration, analysis, and presentation of knowledge from large medical simulation video data collections. Signal processing and machine learning techniques are used to: (1) preprocess audio data, (2) partition audio stream into short segment such that there is only one speaker per segment, (3) train a classifier to recognize each speaker, (4) train a classifier to recognize the emotion of the speaker, and (5) provide statistics that summarize the audio recording. A system that integrates these steps will allow the physician to efficiently retrieve video shots that relate to "**who spoke, when, and the emotion of the speaker**".

Our main contributions can be summarized as follows:

- We propose soft bag-of-word (BoW) feature representations of speech data for speaker identification. In particular, we define three types of BoW that are based on crisp, fuzzy, and possibilistic voting. Instead of working directly in the original spectral feature space, our soft BoW approach maps low-level audio features to more meaningful histogram descriptors. The key advantage of this representation is that speech segments of different lengths will be mapped to feature vectors of equal dimensions. We show that using our mappings with standard classifiers outperform existing methods for speaker and emotion recognition.

- We propose a generalization of the BoW feature representation based on Fisher Vector (FV) for speaker identification. FV uses the Fisher Kernel principle and combines the benefits of generative and discriminative approaches by computing the gradient of the sample log-likelihood with respect to the model parameters.
- We propose two fusion methods for speaker segmentation. We show that our approach can detect more true speaker changing points.
- We adapted a semi-supervised learning algorithm and combined it with the proposed Fisher Vector feature representation to develop a semi-supervised speaker identification method. We show that, when labeled training data is limited, the semi-supervised approach can improve the performance in speaker identification.
- We apply the proposed soft BoW and FV feature representation approaches to emotion recognition. This additional feature provides the physicians the ability to retrieve speech segments from simulation videos based on emotion.
- We develop a graphical user interface (GUI) that combines all of the above features and algorithms. Using this GUI, the physician can efficiently identify who spoke and when. In addition, our system can extract useful statistics and features in a completely unsupervised way. Examples include: for how long

did each person spoke? What is the longest uninterrupted speech segment? Is there an unusual large number of pauses within the speech segment of a given speaker?

The remainder of this dissertation is organized as follows. Chapter 2 provides a review of some algorithms related to speech feature extraction, speaker segmentation, and recognition. Chapter 3 introduces our proposed speaker segmentation algorithms. Chapter 4 introduces our variations of the proposed bag-of-words feature representations. In chapter 5, we introduce our proposed semi-supervised speaker identification method, and in chapter 6 we provide experimental results of the proposed methods, and describe the implemented GUI and its features. Finally, chapter 7 provides conclusions.

CHAPTER 2

RELATED WORK

In this chapter, we first provide an overview of a typical audio/speech data analysis system. Then, we survey some existing algorithms for each component of the system including feature extraction, speaker segmentation, and recognition. Finally, we outline some evaluation and comparison tools.

2.1 Overview of A Typical Speech Data Analysis System

In the last decades, researchers within the speech processing community have proposed several algorithms for speech data analysis such as feature extraction, speaker segmentation, speaker clustering, and speaker recognition, and have used them in various applications [1–11]. In addition to being used in speech processing tasks [12–14], these algorithms have also been used in combination with video data for medical [15] and security [16, 17] applications.

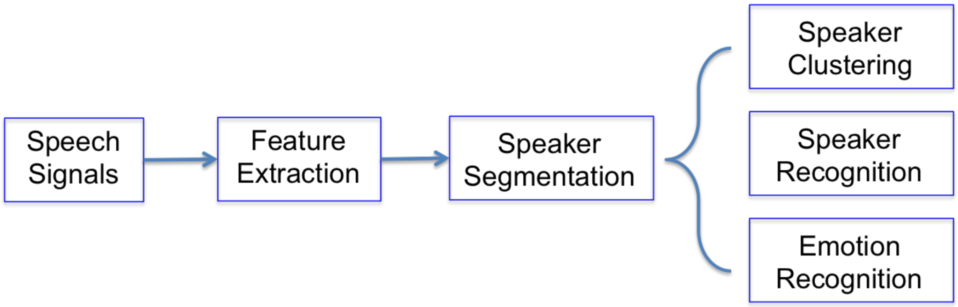


Figure 2.1: Overview of a typical speech data analysis system.

Figure 2.1 illustrates some components that may be included in a typical speech analysis system. Common applications of speech data analysis include:

- **Automatic transcription** [1,18]: This consists of an autonomous system for transcribing radio and television broadcast news based on speech recognition (speech to text) technology. In this application, speaker specific acoustic models are used to improve the transcription accuracy.

- **Audio or audio-visual archiving**: This application involves storing and indexing audio and video content in databases for content-based information retrieval [2,3,19]. In this application, speaker segmentation and feature extraction are two important steps for organizing large audio and video data into semantic categories.

- **Speaker diarization**: The objective of this application is to determine "who spoke and when". Here, first speaker segmentation is used to segment the audio sequence into segments, where each segment corresponds to only one speaker. Then, clustering is used to identify clusters of segments. Ideally, each cluster would include segments of only one speaker. Speaker diarization has also been jointly used in speaker tracking applications [4,5,12–14,20].

- **Audio classification** [21,22]: This application involves classifying utterances into different audio types. Here, audio segmentation is first used to partition the audio sequence into homogeneous segments. Then, a classifier is used to annotate the audio type of each segment based on the extracted features. For instance, an audio segment can be labeled as music, commercial, speech, environmental background noise, or other acoustic types. Furthermore, the speech segments can be classified into different speakers. This task is necessary for effective large vocabulary continuous speech recognition (LVCSR), which includes speaker identification, verification, or tracking [6]. Speech and speaker recognition can also be applied to content spoken document retrieval [10, 11, 23–28].

- **Other applications**: various other applications such as multimedia archive management [2], dialogue detection in movies [29], social network analysis [30], medical assessment (e.g. depression) [31], music information retrieval [32], and audio characterization in security surveillance systems [33] use speaker segmentation and feature extraction algorithms as preprocessing steps, and their performance could be significantly affected by these algorithms.

The rest of this chapter outlines some algorithms for common speech data processing and analysis tools.

2.2 Feature Representation

2.2.1 Feature Extraction

Audio is recorded, stored and represented by a digital waveform with amplitude and sampling frequency. Let $x[n]$ denote the n^{th} sample of the digital wave and let f_s denote the sampling frequency. Usually, the frequency f_s is set to 44.1kHz and one second contains 44100 samples, and n varies from 1 to f_s*(length of audio segment in seconds). Thus, to represent few hours of recorded audio, the length of the signal, $x[n]$, would be too large. Consequently, instead of dealing with the raw signal, audio features that can represent the audio stream by a lower dimensional description that captures the salient features need to be extracted. Some commonly used audio feature extraction methods are presented in the following subsections.

2.2.1.1 Mel-Frequency Cepstral Coefficient (MFCC)

The human ear resolves frequencies non-linearly across the audio spectrum [34]. Initially, Audio Spectrum Envelope Descriptor (ASED), with log-scale bands [35], was used to address this problem. However, the simple rectangular form filters used in ASED do not match the human perception accurately. In [36], the authors introduced the Mel frequency scale, which takes into account how humans perceive the difference between sounds of different frequencies. As a reference point, the pitch of a 1 kHz tone, 40 dB above the perceptual hearing threshold, is defined as 1000 Mels. Other subjective pitch values are obtained by adjusting the frequency of a tone such that it is half or twice the perceived pitch of a reference tone (with a known Mel-frequency). The conversion between Hertz and Mel is given by:

$$m = 2595 \log_{10}(1 + f/700), \tag{2.1}$$

where f is the frequency in Hertz and m is the frequency in Mels. The Mel-frequency coefficients are derived from the short time power spectra, by filtering it with a bank of Mel-scale filters. Fig. 2.2 illustrates the process of the MFCC feature extraction.

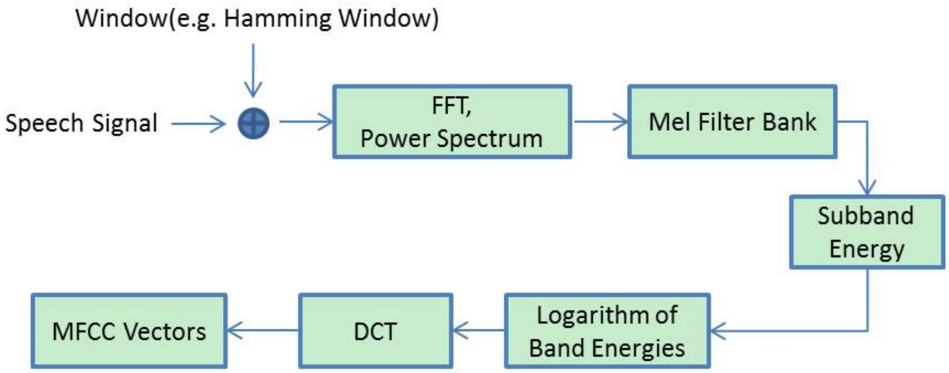


Figure 2.2: Different steps involved in extracting the MFCC features.

The Mel-filter bank amplitudes are highly correlated because of the overlap between adjacent filters as shown in Fig. 2.3. Thus, a cepstral transformation is used to reduce the dimensionality and dependency of the coefficients. Typically, the

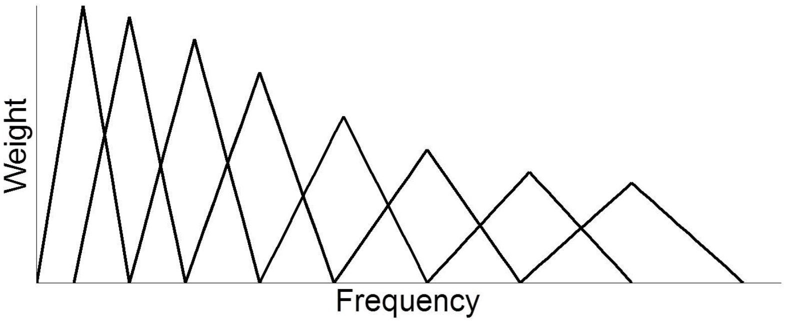


Figure 2.3: A triangular Mel-filter bank.

discrete cosine transformation is applied for this purpose. That is, the coefficients are computed using:

$$c_n = \sqrt{\frac{2}{K}} \sum_{k=1}^{K} (log S_k * cos[n(k-0.5)\pi/K]), n = 1, ..., L \tag{2.2}$$

In (2.2), K is the desired number of sub-bands, and $L \ll K$ is the desired length of the cepstrum, S_k is the kth Mel-scale spectrum value, and c_n is the nth MFCC coefficients.

The MFCC feature vector has been widely used in several applications, such as speech recognition, audio retrieval, and classification [6,11,21,23,26,37–40]. It has proved to be an excellent representation of the human voice and musical signals.

2.2.1.2 Perceptual Linear Prediction (PLP)

The PLP speech analysis technique [41] is based on the short-term spectrum of the signal. Several variations of this representation, using psychophysically based spectral transformations, have been proposed [41,42]. The PLP technique, like most other short-term spectrum based techniques, can be unreliable when the short-term spectral values are modified by the frequency response of the communication channel. Recently, the relative spectra filtering (RASTA) PLP [42] was developed to make the PLP more robust to linear spectral distortions. It is based on the observation that the human speech perception seems to be less sensitive to such steady-state spectral factors. The different steps involved in extracting the RASTA-PLP features are outlined below:

1. Compute the critical-band spectrum by discrete Fourier transform (DFT) and take its logarithm.

2. RASTA processing: estimate the temporal derivative of the log critical-band spectrum, y, using regression line:

$$y = \ln(1 + Jx), \tag{2.3}$$

   where x is the auditory power spectral amplitude, J is a singal-dependent positive constant. The amplitude-warping transform is linear-like for $J \ll 1$ and logarithmic-like for $J \gg 1$.

3. RASTA filtering: reintegrate the log critical-band temporal derivative using a first order infinite impulse response (IIR) system. The whole reintegration process is equivalent to a bandpass filtering of each frequency channel through an IIR filter with a transfer function $H(z)$:

$$H(z) = 0.1z^4 * \frac{2 + z^{-1} - z^{-3} - 2z^{-4}}{1 - 0.98z^{-1}}. \tag{2.4}$$

   The low cut-off frequency of the filter determines the fastest spectral change of the log spectrum, while the high cut-off frequency determines the fastest spectral change that is preserved. In (2.4), the low cut-off frequency is 0.26Hz. The filter slope declines 6 dB/oct from 12.8Hz with sharp zeros at 28.9 and at 50Hz.

4. Transform the filtered speech representation through expanding static nonlinear transformation and add the equal loudness curve and multiply by 0.33 to

simulate the power law of hearing:

$$\Phi(\omega) = (E(\omega)\Omega(\omega))^{0.33} \tag{2.5}$$

where $\Omega(\omega)$ is the IIR filtered spectrum, $\Phi(\omega)$ is the power law of hearing, $E(\omega)$ is a nonlinear transfer function with equal loudness curve and is defined as:

$$E(\omega) = \frac{(\omega^2 + 56.8 * 10^6)\omega^4}{(\omega^2 + 6.3 * 10^6)^2 * (\omega^2 + 0.38 * 10^9)}. \tag{2.6}$$

The inverse of equation (2.3) is:

$$x = \frac{e^y - 1}{J}. \tag{2.7}$$

To ensure the positivity of the processed power spectrum, the inverse transform in step (5) is approximated by:

$$x = \frac{e^y}{J}. \tag{2.8}$$

5. Take the inverse logarithm of this relative log spectrum followed by inverse discrete Fourier transform (IDFT), to obtain a relative auditory spectrum.

6. Compute an all-pole model of this spectrum by using Levinson-Durbin recursion algorithm [43] to obtain RASTA-PLP cepstral coefficients.

Fig. 2.4 illustrates the flowchart of the extraction of RASTA-PLP features.

Similar to the MFCC features, the PLP has also been widely used in audio clustering and classification [24, 44]. It has proved to be an excellent representation of the human voice and musical signals.

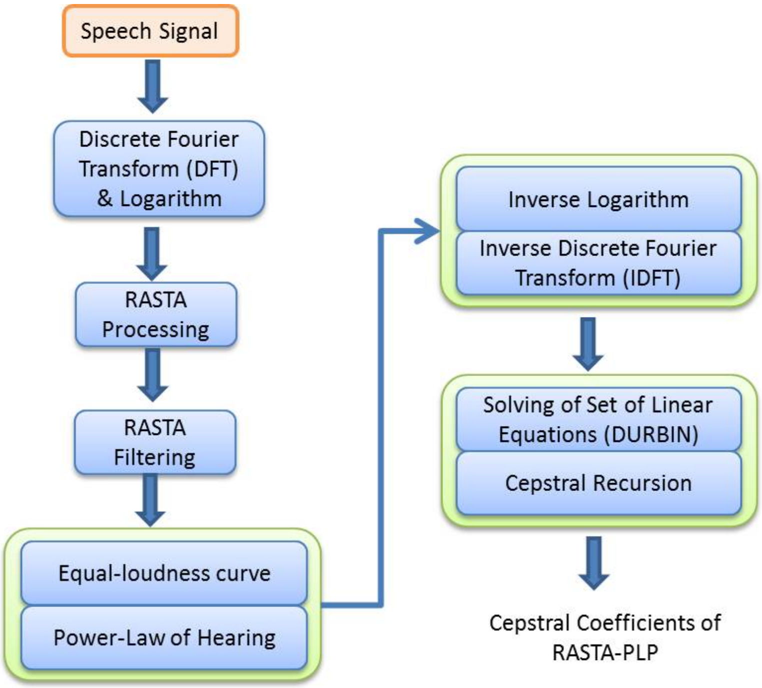

Speech Signal
Discrete Fourier Transform (DFT) & Logarithm
RASTA Processing
RASTA Filtering
Equal-loudness curve
Power-Law of Hearing
Inverse Logarithm
Inverse Discrete Fourier Transform (IDFT)
Solving of Set of Linear Equations (DURBIN)
Cepstral Recursion
Cepstral Coefficients of RASTA-PLP

Figure 2.4: Different steps involved in the extraction of RASTA-PLP features.

2.2.1.3 Short-Time Energy, Zero Crossing Rate, and Spectral Centroid

Typically, speech is a slowly varying signal, changing every 50-100ms. Thus, it is common to process it in frames of about 10ms, during which the speech waveform can be considered stable.

The short-time energy (STE) is defined as:

$$E_m = \sum_{n=-\infty}^{\infty} (x[n]w[m-n])^2 \tag{2.9}$$

where $w[n]$ is the Hamming window with 25ms width and 10ms sliding, E_m is the short-time energy at the m^{th} Hamming window. The Hamming window is defined as:

$$w[n] = \begin{cases} 0.54 + 0.46cos(\frac{2\pi n}{N-1}) & 0 \leq n \leq N-1 \\ 0 & \text{otherwise} \end{cases} \tag{2.10}$$

where N is the Hamming window width.

The zero crossing rate (ZCR) is another feature commonly-used in characterizing audio signals. It is computed by counting the number of times the audio waveform crosses the zero axis, and normalizing it by the length of the input signal $x[n]$ [45]. Formally, the ZCR is defined as:

$$ZCR = \frac{1}{2}(\sum_{n=1}^{N-1} |sign(x[n]) - sign(x[n-1])|)\frac{fs}{N}, \tag{2.11}$$

where N is the number of samples in $x[n]$, fs is the sampling frequency, and $sign(x)$ is the signum function defined as:

$$sign(x) = \begin{cases} 1 & \text{if } x > 0 \\ 0 & \text{if } x = 0 \\ -1 & \text{if } x < 0 \end{cases} \tag{2.12}$$

The spectral centroid (SC) characterizes a spectrum of a digital signal. It indicates where the "center of mass" of the spectrum is [46]. Perceptually, SC has a robust connection with the impression of "brightness" of a sound. SC is calculated as the weighted mean of the frequencies present in the signal, determined using a Fourier transform, with their magnitudes as the weights:

$$SC = \frac{\sum_{n=0}^{N-1} c(n)f(n)}{\sum_{n=0}^{N-1} f(n)}. \tag{2.13}$$

In (2.13), $f(n)$ represents the weighted frequency value of bin number n, and $c(n)$ denotes the center frequency of that bin.

The STE, ZCR, and SC features are effective for distinguishing between voiced and unvoiced speech regions. Voiced regions have higher energies and lower zero crossing rates than unvoiced regions.

2.2.2 Feature Preprocessing

For many audio signals, especially speech streams, there exist silence sections with various lengths. Usually, most segmentation algorithms segment the silence into heterogeneous sub-segments and thus, increase the possibility of false alarms.

Due to background noise, signal interruption, or some other noise, the energy of a silence window is not necessarily zero. Some methods use a background energy threshold to detect and remove silence [47]. The first step in these methods is to compute the Hamming energy for all frames and then get a sequence $(f^1, f^2, ..., f^{N_r}, ...f^n)$, where f^i represents the Hamming energy in the ith frame. Then, a threshold of the Hamming energy is defined using:

$$Threshold = \frac{1}{N_r}\sum_{i=1}^{N_r} E(f^i). \tag{2.14}$$

In (2.14), $E(f^i)$ denotes the Hamming energy in frame f^i, N_r denotes the number of frames with the highest r percentage of Hamming energy. All values lower than this threshold are considered as silence. After determining the threshold, all silence features would be removed from the audio stream. The remaining speech features are used to perform the segmentation procedure.

2.3 Speaker Segmentation

The architecture of a typical speaker segmentation system is illustrated in Fig. 2.5. It has five main components: feature representation, feature preprocessing, feature modeling, change point detection, and adjacent speech segment merging. These components are described in the following subsections.

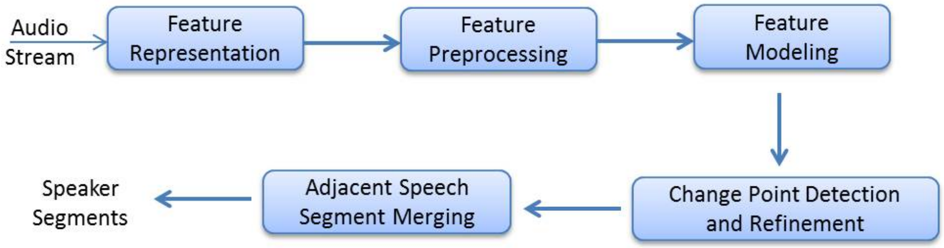


Figure 2.5: Architecture of a typical speaker segmentation system.

2.3.1 Segmentation Algorithms

Previous work on speaker segmentation has focused on four main approaches: decoder-guided [48], model-based [49], metric-based [50,51], and information criterion-based [52–54].

In decoder-guided audio segmentation, first the input audio stream is decoded. Then, the stream is cut at the silence locations that generated from the decoder to produce the desired segments. Other information from the decoder, such as gender information, could also be used for the segmentation. The limitation of this method is that it can only detect change points at silence locations, which generally are not directly connected with the acoustic changes of the speech signals [48].

In model-based segmentation methods, first a set of models is trained for different speakers. Then, a new speech segment is classified according to how it fits the trained models. Various methods have been used to create training models. For instance, the universal background model (UBM) [55] is trained by using a large volume of speech data offline. An extension of the UBM uses two universal gender models (UGM) that can discriminate between male and female speakers instead of just one model [55]. Another model-based approach, called anchor model, projects speaker segments into a subspace of reference speakers [56]. The sample speaker model (SSM) [55] learns a general speaker-independent model. Then it adapts the general model to each speaker to learn speaker-dependent models. In addition to the above methods, several model-based segmentation algorithms based on hidden Markov models (HMMs) [1,57,58] or support vector machines (SVMs) [59] have been proposed.

The third approach to speaker segmentation is metric-based. In this approach, the audio stream is segmented by detecting the local minimum of a proper distance between neighboring windows. Various metric based algorithms have been proposed. Some of these are based on Kullback-Leibler divergence (KL or KL2) [52,55], and the generalized likelihood ratio (GLR) [1, 55]. Others, are based on the Bhattacharyya distance [60], or a weighted squared Euclidean distance, which updates the weights by Fisher linear discriminant analysis [61].

Information criterion-based segmentation methods are similar to the GLR approach. They also consider the penalty in the segmentation procedure. The Delta Bayesian information criterion (ΔBIC) [53] is an example of such approach. It is threshold-free, which makes it suitable for unknown acoustic conditions. The ΔBIC based segmentation algorithm has been used in window-growing-based segmentation (WinGrow) [53,62,63], fixed-size sliding window segmentation (FixSlid) [52,60,64–66], two-pass distance-based BIC (DISTBIC) [67], and cross probabilities method (XBIC) [68].

Some researchers use hybrid algorithms, where they first use metric-based segmentation to create an initial set of speaker models, and then apply model-based techniques to refine the segmentation [57]. Another hybrid system, where the audio stream is recursively divided into two sub-segments and speaker segmentation is applied to each segment independently, was proposed in [69]. In [1], two systems, LIA (Laboratoire informatique d'Avignon) based on HMM and CLIPS (Communication Langagiere et Interaction Personne-Systeme) based on BIC speaker segmentation, are combined and followed by hierarchical clustering.

2.3.1.1 Chen's Bayesian Information Criteria (*ChenBIC*) Based Over-segmentation

In [53], the BIC algorithm was applied to perform speaker segmentation and clustering. This approach models the input audio stream as a Gaussian process in the cepstral space, and uses the maximum likelihood approach to detect turns of the Gaussian process. The decision of a turn is based on the BIC. The BIC was also used as a termination criterion in the hierarchical merging of audio segments. In other words, two nodes can be merged only if the merging operation increases the BIC value.

As a model selection criterion, the ΔBIC is widely used in speaker segmentation [53, 54, 56]. It is defined as:

$$\Delta BIC = GLR - P, \tag{2.15}$$

where

$$GLR = \log \frac{Pr(\mathbf{X}|\boldsymbol{\mu}_x, \boldsymbol{\Sigma}_x) Pr(\mathbf{Y}|\boldsymbol{\mu}_y, \boldsymbol{\Sigma}_y)}{Pr(\mathbf{Z}|\boldsymbol{\mu}_z, \boldsymbol{\Sigma}_z)} \tag{2.16}$$

is the general likelihood ratio, and

$$P = \frac{1}{2}\lambda(d + \frac{1}{2}d(d+1)) \log N \tag{2.17}$$

is a penalty term that reflects the model complexity. In (2.16), $\mathbf{X} = \{\boldsymbol{x}_i \in R^d, i = 1, ..., N_x\}$ and $\mathbf{Y} = \{\boldsymbol{y}_j \in R^d, j = 1, ..., N_y\}$ are feature vectors from two utterances, and each one is modeled by a Gaussian distribution, i.e. $\mathbf{X} \sim N(\boldsymbol{\mu}_x, \boldsymbol{\Sigma}_x)$ and $\mathbf{Y} \sim N(\boldsymbol{\mu}_y, \boldsymbol{\Sigma}_y)$. Similarly, $\mathbf{Z} = \mathbf{X} \cup \mathbf{Y}$, is modeled by a Gaussian, and $\mathbf{Z} \sim N(\boldsymbol{\mu}_z, \boldsymbol{\Sigma}_z)$. In (2.17), d is the dimensionality of one feature vector, $N = N_x + N_y$ is the number of feature vectors (or audio frames) in $\mathbf{Z}$.

The ΔBIC in (2.15) is basically a thresholding of the GLR with penalty P. It could be viewed as a distance measure between two clusters. If two clusters, $\mathbf{X}$ and $\mathbf{Y}$, are similar and if merged into one cluster, $\mathbf{Z}$, they could be approximated by one Gaussian component, then, they are considered similar. The advantage of using ΔBIC as a distance is that the appropriate threshold could be easily designed by adjusting the penalty factor λ.

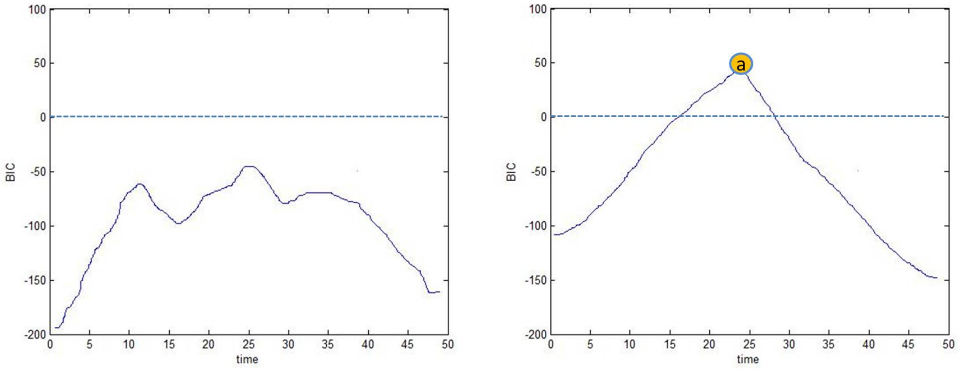
While applying the ΔBIC criterion to **d**etect **o**ne speaker **c**hanging point (DOC) in the analysis window $[a, b]$ (i.e. start from a to b), all ΔBIC values at time t, for $a < t < b$, are computed using equation (2.15), where $\mathbf{X}$ is the segment from window $[a, t]$, and $\mathbf{Y}$ is the segment from window $[t, b]$. If the maximum of ΔBIC values, which is located at t', is larger than zero, then t' is detected as one speaker changing point. Fig. 2.6 (a) shows an example with all $\Delta BIC < 0$ and no change point detected, and Fig. 2.6 (b) shows an example with $\Delta BIC > 0$ and one speaker change point detected at a location.

The basic *ChenBIC* algorithm [53] that is used to detect multiple changing points is outlined below:

Fig. 2.7 shows the details of *ChenBIC* speaker segmentation method for an

Algorithm 2.1 Speaker Segmentation by *ChenBIC* algorithm

```
Initialize the interval window $W_{ini} = [a, b]$
repeat
    Detect whether there is one changing point in window $[a, b]$ via the $DOC$ method
    if no change in $[a, b]$ then
        let $b = b + W_g$, where $W_g$ denotes the length of the window growing
    else
        let t be the detected changing point and set $a = t, b = a + W_{ini}$
    end if
until Reach the end of the audio stream
return changing points
```



(a) (b)

Figure 2.6: BIC curves of two uterances with: (a) no changing point, (b) one changing point.

audio stream with 3 speakers denoted $Seg1$, $Seg2$, and $Seg3$. The goal is to detect the change points P and Q. *ChenBIC* starts from W_{ini}, usually set to 2s ($a = 0$, $b = 2$). If no changes in the analysis window, the window size is grown by W_g. When P is detected as a changing point, the window is slid to start from position P with initial window size W_{ini} and detect the next point. When the window size reaches the maximum size W_{max}, the whole analysis window will shift by W_g seconds. This process if repeated until the next changing point Q is detected. Then, the algorithm is reset to start from Q and detect the rest of the changing points in the audio.

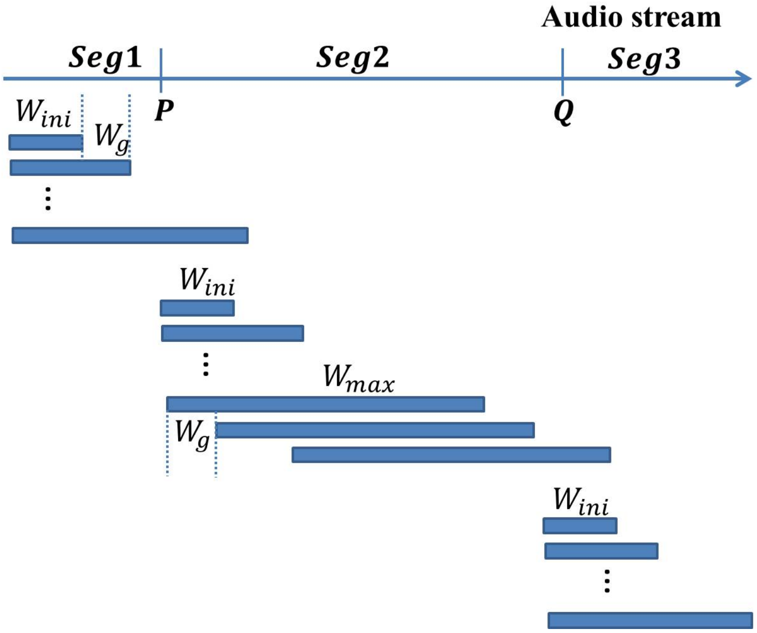


Figure 2.7: Diagram of $WinGrow$ for speaker change detection.

The $ChenBIC$ algorithm detects most change points even if they are not significant, and typically, generates over-segmented results. It has quadratic complexity that can be reduced by crude search without sacrificing the resolution.

2.3.1.2 Cheng's Sequential Metric-based Segmentation ($SeqBIC$)

In [70], Cheng proposed a template-based multiple changing points' detection method, in which each change point has multiple chances to be detected by different window sequences. At the beginning, the initial window is set to 12 seconds, when the BIC value becomes positive at time t_j, then the change point is refined by performing BIC and relocated with the maximum BIC value at the range of $[t_j - 2, t_j + 2]$s. Then, by shifting the analysis window, all possible change points are detected sequentially. The details of the $SeqBIC$ algorithm are outlined in Algorithm 2.2.

Algorithm 2.2 Speaker Segmentation by *SeqBIC* algorithm

```
Set the initial window $W_{ini} = [a, b], a = 0, b = 12$
repeat
    Use a 2-second window of MFCC audio features as the template
    Detect the first changing point in $W_{ini}$ by template-based BIC method
    if no changing point is detected then
        Change the template length to 3 seconds
        Detect the first changing point in $W_{ini}$ by template-based BIC method
    end if
    if no changing point is detected in previous step then
        Shift the window by 2 seconds, i.e. $a = a + 2, b = a + 12, W_{ini} = [a, b]$
    else
        Let $t$ be the detected changing point in $W_{ini}$
        Shift the window to $t$, i.e. $a = t, b = a + 12$
    end if
until all possible changing points are detected
return changing points
```

In *SeqBIC*, each speaker change point has multiple chances to be detected. After ΔBIC curve is obtained, BIC-based algorithm is performed at $t_j \pm 2$ s to locate the exact change point, and merge the false alarms.

2.3.1.3 Divided-and-Conquer (DAC) Strategies Based Speaker Segmentation

Recently, Cheng [71] developed three BIC-based speaker segmentation algorithms based on DAC strategies for detecting multiple change points in one analysis window. All are modifications of Chen's [53] BIC algorithm. These three methods, $DAC1, DAC2$, and $DAC3$, assume that feature vectors from the different speakers have different Gaussian distributions. These three algorithms are outlined below:

In general, when data samples are derived from more than one Gaussian distribution, two Gaussians fit the distribution of the data better than one Gaussian.

Algorithm 2.3 Speaker Segmentation by $DAC1$ algorithm

```
Set the initial window $W_{ini} = [a, b], a = 0, b = 12$
Detect a changing point in the analysis window $W$ using Chen's BIC over-segmentation method
if no changing points are detected in $W$ or the size of $W$ is smaller than the minimum length then
    return changing points $CP$
else
    repeat
        (Divide Part) let $t$ be the change point detected in step 1, divide $W$ into two sub-windows, $W1$ and $W2$, at $t$
        Recursively compute $CP_{W1} \leftarrow DAC1(W1)$; and $CP_{W2} \leftarrow DAC1(W2)$
        (Combine Part) set $CP = t \bigcup CP_{W1} \bigcup CP_{W2}$
    until no changes detected
end if
return changing points $CP$
```

$DAC1$ can always detect the changing points while ΔBIC is positive. However, if two or more segments in the analysis window are derived from the same speaker, the performance of $DAC1$ declines. This is due to the fact that λ value in equation (2.17) is variable for different audio stream and should be setup manually.

$DAC2$ overcomes the limitation caused by the difficulty of determining λ in $DAC1$.

Algorithm 2.4 Speaker Segmentation by $DAC2$ algorithm

```
1: If the size of the analysis window $W$ is smaller than minimum length then stop and return; otherwise continue to step 2
2: (Divide Part) use Chen's BIC method in $W$, let $t$ be the time with largest $\Delta$BIC value; and divide W into two sub-windows, $W1$ and $W2$, at $t$
3: Recursively compute $CP_{W1} \leftarrow DAC2(W1)$; and $CP_{W2} \leftarrow DAC2(W2)$
4: (Combine Part) if $\Delta$BIC value in $W1$ and $W2$ at time $t$ in step 2 is positive, then set $CP = t \bigcup CP_{W1} \bigcup CP_{W2}$; otherwise let $X$ be the segment on the left of $t$ in $W1$ and $Y$ be the segment on the right of t in $W2$, and if $\Delta$BIC value of $X \bigcup Y$ at t is positive, then set $CP = t \bigcup CP_{W1} \bigcup CP_{W2}$, else it is not a change point, and merge $X$ and $Y$
5: return changing points $CP$
```

$DAC3$ is developed based on $FixSlid$ with GLR distance measurement method instead of Chen's BIC algorithm.

Algorithm 2.5 Speaker Segmentation by $DAC3$ algorithm

```
1: Initially obtain $DP_{set} = \{DP_1, ..., DP_N\}$ to be divide-points in W obtained from $FixSlid$ [52] with $GLR$ distance method; $GLR_{set} = \{GLR_1, ..., GLR_N\}$ and $GLR_i$ is $GLR$ value at $DP_i$
2: If $DP_{set}$ is empty, then stop and return; otherwise go to step 3
3: ($Divide\ Part$) let $DP_k$ be the point with maximum value in $GLR_{set}$, and let $t$ be the time index of $DP_k$, divide $W$ into two sub-windows, $W1$ and $W2$, at $t$; and then divide $DP_{set}$ into two sub-sets, $DP_{set1} = \{DP_1, ..., DP_{k-1}\}$, and $DP_{set2} = \{DP_{k+1}, ..., DP_N\}$
4: Recursively compute $CP_{W1} \leftarrow DAC3(W1, DP_{set1}, GLR_{set1})$; and $CP_{W2} \leftarrow DAC3(W2, DP_{set2}, GLR_{set2})$
5: ($Combine\ Part$) let $X$ be the segment on the left of $t$ in $W1$ and $Y$ be the segment on the right of $t$ in $W2$, and if $\Delta$BIC value of $X \bigcup Y$ at $t$ is positive, then set $CP = t \bigcup CP_{W1} \bigcup CP_{W2}$, else it's not a change point, and merge $X$ and $Y$
6: return changing points $CP$
```

The major difference between $DAC2$ and $DAC3$ is that $DAC2$ detects changing points by Chen's method in the *Divide* stage, and only the divide-points with negative values calculated in the *Divide* stage are verified by segment merging based on the values of their neighboring segments in the $Combine$ stage. In contrast, $DAC3$ detects change points by verifying all the input divide-points indicated by $FixSlid$ using segment merging. Both $DAC2$ and $DAC3$ find the change point in the $Combine$ stage.

In Fig. 2.8, the recursive tree for the DAC algorithms is illustrated. Each tree node represents a divide point; the number inside the node indicates the order of the division, while the number below the node indicates the order in *Combine* step. C_2 is first detected as a changing point, and then recursively detecting the changing points on the left and the right of C_2, and finally eight changing points detected, including real ones at C_1 and C_2, and other six false alarms.

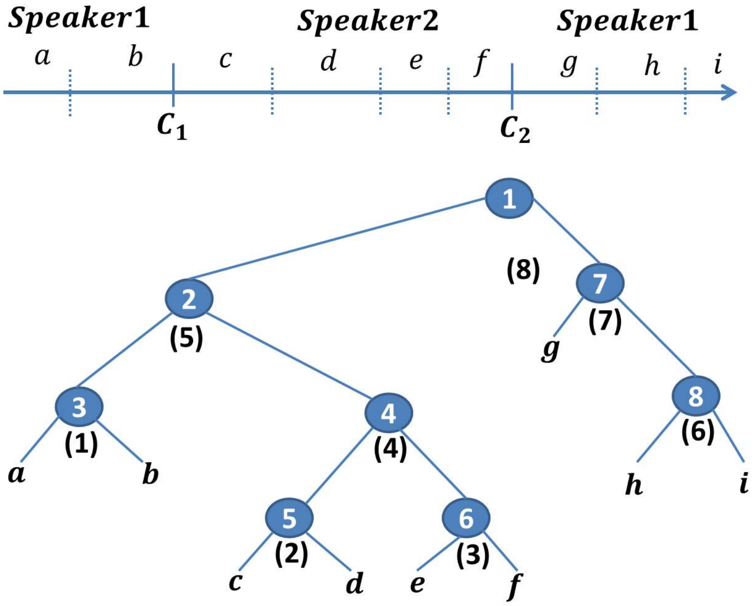


Figure 2.8: Recursive tree that simulates the recursive process of the DAC algorithms.

2.3.1.4 Other Typical BIC-based Segmentation

In [67], a two-step segmentation technique, called distance-based BIC (DIST-BIC), was proposed. First, a distance is used to determine potential speaker changing points. Then, BIC is used to discard less likely changing points. Six metrics were applied in this first step: GLR, Kullback-Leibler divergence, and four similarity measures derived from second-order statistics. In the second step, a BIC based validation algorithm is applied on the local maxima of these metrics. A local maximum is considered to be significant if:

$$\begin{cases} |d(max) - d(min_r)| > t_d\sigma \\ |d(max) - d(min_l)| > t_d\sigma \end{cases} \tag{2.18}$$

In (2.18), d is the distance, σ denotes the standard deviation of the distances d, t_d is a threshold, and min_r and min_l are minimum to the left and to the right of the maximum in d, $d(max)$.

In [47], Wu and Hsieh proposed an algorithm that can detect multiple speaker change points in one analysis window. First, silent parts are detected and deleted. Then, the minimum description length (MDL) is used instead of the BIC to detect change points. This approach uses multiple sets of features including the 12-order MFCC and their corresponding first-order differences, the logarithmic energy, and the first-order difference of the logarithmic energy.

A different unsupervised speaker segmentation approach that uses the Hotelling T^2 statistic to pre-select candidate speaker change points was proposed in [62]. These candidate points are then evaluated using the BIC. This approach also relies on a variable-size window and frame skipping. For features, this algorithm uses frame energy, 12-order MFCC, and their first order differences. Combining the Hotelling T^2 and BIC have two main advantages: First, the two-stage processing reduces computation complexity; Second, it guarantees that the segments are not too short, and are sufficient to estimate the model parameters.

Another interesting speaker segmentation algorithm was proposed in [57]. This algorithm is hybrid and includes three main steps: First, T^2 statistics are computed in a template window and a potential change point is detected by maximizing the statistics. Each candidate change point is validated by the BIC. Second, hierarchical clustering is performed by merging segments according to the difference of their BIC values. Third, hidden Markov model (HMM) is performed on each cluster to estimate its model parameters. In [57], the authors showed that the hybrid algorithm is more efficient than the metric-based algorithm.

Most current speaker segmentation algorithms suffer from the following limitations. First, the audio stream used for segmentation should include only one speaker at a time. If multiple speakers are speaking simultaneously, most algorithms would be confused and cannot provide reasonable segmentation. Second, for window-based algorithms, choosing the optimal template analysis window size is not trivial and can have a significant effect. If the window size is too large, it may contain multiple speaker changes. This would cause misdetection. On the other hand, if the size is too small, it may not include enough features to obtain reasonable estimates of the model parameters. Moreover, inaccurate model parameters would not allow the merging of adjacent windows in the refinement stage.

2.3.2 Segment Feature Representation

2.3.2.1 Gaussian Mixture Model (GMM)

A GMM is a mixture of several Gaussian distributions and is used to estimate the Probability Density Function (PDF) of a set of feature vectors. Given the observations, the likelihood of a GMM is then represented as:

$$p(\boldsymbol{x}|\lambda) = \sum_{i=1}^{M} \omega_i N(\boldsymbol{x}|\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i) \tag{2.19}$$

where $\boldsymbol{x}$ is a D-dimensional feature vector, M is the number of Gaussian components in the GMM, ω_i is the ith mixture weight satisfying the constraint $\sum_{i=1}^{M} \omega_i = 1$, $0 \leq \omega_i \leq 1$, and $N(\boldsymbol{x}|\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)$ is a multivariate Gaussian probability density function:

$$N(\boldsymbol{x}|\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i) = \frac{exp\{-\frac{1}{2}(\boldsymbol{x} - \boldsymbol{\mu}_i)^T \boldsymbol{\Sigma}_i^{-1} (\boldsymbol{x} - \boldsymbol{\mu}_i)\}}{(2\pi)^{D/2} |\boldsymbol{\Sigma}_i|^{1/2}}. \tag{2.20}$$

In (2.20), μ_i is the mean vector and Σ_i is the covariance matrix of the i^{th} component. The parameters of a GMM $\lambda = \{\omega_i, \boldsymbol{\mu}_i, \boldsymbol{\sigma}_i\}_{i=1}^{M}$ can be learned through the well-known expectation-maximization (EM) algorithm [72] by maximizing the likelihood of the data.

Generally, the N observations, $\mathbf{X} = \{\boldsymbol{x}_1, ..., \boldsymbol{x}_N\}$, where $\boldsymbol{x_i}$ is ith feature vector, are assumed to be independent and identically distributed (*i.i.d.*). Thus, the likelihood of a GMM parameterized by λ given $\mathbf{X}$ can be estimated using:

$$P(\mathbf{X}|\lambda) = \prod_{n=1}^{N} p(\boldsymbol{x}_n|\lambda). \tag{2.21}$$

The GMM has been extensively used for acoustic/speaker modeling, especially in text-independent speaker recognition applications [7, 39]. The GMM has several properties that motivate its use for modeling individual speakers [73]. In particular,

1. GMM can be viewed as a probabilistic modeling of speaker dependent acoustic classes with each Gaussian component corresponding to an acoustic class, such as vowels, nasals, and fricatives etc.

2. GMM can approximate arbitrarily shaped densities using a finite number of Gaussian basis functions.

2.3.2.2 GMM Adaptation

The universal background model (UBM) is an approach that uses all available training data to learn the parameters of a single model [7, 74]. It is trained using a considerable amount of speech data from a set of speakers. Thus, it represents the main characteristics of the global speech signals. The UBM consists of a mixture of M GMM, as defined in (4.15).

Initially, a single speaker-independent UBM, λ_0, is trained using all utterances from all speakers in the training set. Then, for a particular utterance from a particular speaker, a GMM λ is derived by updating λ_0 using the maximum a posteriori (MAP) procedure [7] outlined below:

1. Given a UBM $\lambda_0 = \{\omega_i^o, \boldsymbol{\mu}_i^o, \boldsymbol{\sigma}_i^o\}_{i=1}^M$, and a training utterance $\mathbf{X} = \{\boldsymbol{x}_t\}_{t=1}^T$, where $\boldsymbol{x}_t$ is a feature vector, and T is the number of feature vectors

2. Determine the probabilistic alignment of the training feature vectors into the UBM's component densities via Bayes rule:

$$p(i|\boldsymbol{x}_t) = \frac{\omega_i^0 N(\boldsymbol{x}_t|\boldsymbol{\mu}_i^0, \boldsymbol{\sigma}_i^0)}{\sum_{j=1}^M \omega_j^0 N(\boldsymbol{x}_t|\boldsymbol{\mu}_i^0, \boldsymbol{\sigma}_i^0)}, \quad for\ i = 1, ..., M,\ and\ t = 1, ..., T. \tag{2.22}$$

   In (2.22), $p(i|\boldsymbol{x}_t)$ is the posterior probability of assigning a training feature vector $\boldsymbol{x}_t$ to the ith mixture component of the UBM λ_0.

3. Compute the sufficient statistics:

$$n_i = \sum_{t=1}^T p(i|\boldsymbol{x}_t), \tag{2.23}$$

$$\boldsymbol{E}_i = \frac{1}{n_i} \sum_{t=1}^T p(i|\boldsymbol{x}_t)\boldsymbol{x}_t, \tag{2.24}$$

   and

$$\boldsymbol{E}_i^2 = \frac{1}{n_i} \sum_{t=1}^T p(i|\boldsymbol{x}_t)\boldsymbol{x}_t^2. \tag{2.25}$$

4. Compute $\lambda_1 = \{\omega_i^1, \boldsymbol{\mu}_i^1, \boldsymbol{\sigma}_i^1\}_{i=1}^M$ that models the training utterance $\mathbf{X}$ using:

$$\omega_i^1 = [\frac{\alpha_i n_i}{T} + (1 - \alpha_i)\omega_i]\tau, \tag{2.26}$$

$$\boldsymbol{\mu}_i^1 = \beta_i \boldsymbol{E}_i + (1 - \beta_i)\boldsymbol{\mu}_i^0, \tag{2.27}$$

   and

$$(\boldsymbol{\sigma}_i^1)^2 = \gamma_i \boldsymbol{E}_i^2 + (1 - \gamma_i)((\boldsymbol{\sigma}_i^0)^2 + (\boldsymbol{\mu}_i^0)^2) - (\boldsymbol{\mu}_i^1)^2 \tag{2.28}$$

where τ is a scaling factor computed over all adapted mixture weights to ensure that they sum to one. The adaptation coefficients, $\{\alpha, \beta, \gamma\}$, used in (2.26)-(2.28) are computed using the empirical formula:

$$v_i = \frac{n_i}{n_i + r^v}, \tag{2.29}$$

where $v \in \{\alpha, \beta, \gamma\}$, and r^v is a fixed relevance factor for v.

2.3.3 Distance Measures

In general, hybrid stages could increase the reliability, robustness, and usability of a speaker segmentation system. Typically, these methods use a coarse pre-segmentation with relatively large windows. Then, they use a posterior processing (refinement) stage to reduce the number of false alarms in the final segmentation. Many postprocessing algorithms for merging adjacent similar speech segments have been proposed. Some of these methods are based on BIC [47, 53, 62, 67]. Others, are based on distance measures. Some of the typically used distances include Euclidean distance, Minkowski distance, Earth Mover Distance (EMD) [75], and diffusion distance [76].

2.4 Speaker Recognition

General speech related recognition tasks involves three main categories: speech recognition, speaker recognition, and language identification. Speech recognition [6], also called speech to text (STT), converts spoken words to text. The second category, speaker recognition [77] consists of validating the user's identity through the characteristics of his/her voice. In other words, speech recognition involves the recognition of what is being said, and speaker recognition is the recognition of who is speaking.

These two categories are sometimes referred to as voice recognition. The third category, language identification, consists of discriminating between natural language and speech content [78].

Speaker recognition encompasses both verification and identification [77]. Speaker verification is to verify a person's claimed identity from his/her voice. This is also known as voice verification, speaker authentication, or talker verification. Speaker identification, on the other hand, is to decide who the person is, or if the person is unknown (in the open-set case). In our work, we focus on the speaker recognition (or identification) task.

In the past decades, researchers have developed various methods for speaker recognition, with a focus on algorithms for speaker modeling and classification. In particular, Support vector machines (SVM) has been widely applied to speaker recognition tasks, and various kernel methods have been proposed. For instance, Fisher Kernels [79], GMM supervector kernels [80], MLLR kernels [81], and cluster adaptive training (CAT) kernels [82] have been proposed to map variable length speech segments into a fixed dimensional representation for the purpose of classification. An alternative approach, based on a logistic regression to train a suitable weighting for each score for classification, was proposed in [83]. In [84], Longworth developed a multiple kernel learning algorithm based on combining derivative and parametric kernels for speaker verification. SVM kernel has also been used in [37], where a GMM-supervector is used to characterize each speaker, and a GMM-UBM mean interval is used to derive the GMM-UBM mean interval (GUMI) kernel and combine it with SVM for speaker recognition. This approach uses a Bhattacharyya based GMM distance that combines both mean and covariance statistical dissimilarities.

In [85], Kinnunen showed that using a standard discriminative classifier (GLDS-SVM) in speaker verification, the GMM-UBM model is suitable for short segments, while the vector quantization based UBM is suitable for long utterances. An auditory based feature extraction algorithm for speaker identification was proposed in [38]. This feature is based on time-frequency transformation and a set of modules to simulate the signal processing functions in cochlear filter bank. In [86], Wang proposed combining MFCC features and phase information for speaker identification. This system selects feature frames and integrates them with mutual information for speaker recognition. Feature frames are determined by the minimum-redundancy within selected feature frames and their maximum-relevancy to the speaker models.

Speaker adaptation methods have also been widely used for speaker verification and identification. For instance, Reynolds proposed a GMM method for speaker recognition and a GMM adaptation, based on the UBM approach, for speaker verification [7, 39]. Various speaker adaptation methods, such as maximum a posteriori (MAP), maximum likelihood linear regression (MLLR), and constrained MLLR for SVM based speaker recognition were compared in [8]. In [22], Zhu proposed the feature space maximum a posteriori linear regression (fMAPLR) and an SVM based classification for speaker verification.

2.5 Bag-of-words Feature Representation

The bag-of-words model has been widely used in various applications, such as document classification, computer vision, speech and speaker recognition. In document classification, the feature is constructed based on the frequency of occurrence of each word [87]. Generally, there are two different models to represent the document. One model uses a vector of binary attributes to indicate whether a word occurs or does not occur in the document. This representation can be modeled as a multivariate Bernoulli distribution. Another model takes the number of word occurrences into account, and represents the document by a sparse histogram of words frequencies. This representation can be modeled as a multinomial model. For both models, the Naive Bayes classifier is commonly used for classification.

In computer vision, a bag of *visual* words is a vector of frequency counts of a vocabulary of local image features. It has been used mainly in image/video scenes classification and retrieval [88, 89]. In [88], a "bag of key points" method was proposed based on vector quantization of affine invariant descriptors of image patches. Two different classifiers, Naive Bayes and SVM, were applied for semantic visual categories classification. Similarly, in [89], a set of viewpoint invariant region descriptors were extracted to search and localize all the occurrences of a given query object in a video. In this approach, a visual vocabulary was built through vector quantizing the descriptors into clusters. Using standard indexing methods used in text retrieval, the term frequency-inverse document frequency (TF-IDF) was computed and the cosine similarity was used for retrieval.

The BoW has also been used for the analysis of speech data. In [90], the high-frequency keywords (e.g. *you know*, *um*, *right*, etc.) were selected by computing the frequent, reflexive words and word pairs, and modeling them via word-based HMM models. Integrating this advantage of text-dependent modeling into the traditional GMM-based text-independent speaker recognition was shown to improve the performance. In [91], a bag-of-words (BoW)-style feature representation, which quantizes the observed direction of arrival (DOA) powers into discrete “word” samples, was developed to solve the speaker-clustering problem. In this approach, a time-varying probabilistic model was combined with the DOA information calculated from a microphone array to estimate the number and locations of the speakers.

Fisher Vector (FV) feature representation [92] is a generalization of the bag-of-word approach, it was shown to achieve great performance in image classification [92–95]. It is based on the Fisher Kernel principle [79]. Fisher kernel combines the benefits of generative and discriminative approaches by computing the gradient of the sample log-likelihood with respect to the model parameters.

2.6 Emotion Recognition

Emotion recognition is to recognize and interpret human emotions. It is an interdisciplinary field spanning computer sciences, psychology, and cognitive science. The motivation for this research is to simulate empathy, the machine should interpret the emotional state of humans and give an appropriate response for these emotions.

Generally, a video recording might contain facial expressions, body posture and gestures, speech, while other sensors detect emotional cues by directly measuring physiological data, such as galvanic skin response, blood volume pulse, and facial electromyography. It would be very useful for emotion recognition by extract meaningful patterns from these different types of data. Also, some useful techniques can be applied, e.g. speech recognition, natural language processing, facial expression detection, etc.

Usually, emotion/affect can be described by psychologists in terms of discrete categories [96], which include *happiness*, *sadness*, *fear*, *anger*, *disgust*, and *surprise*. The main advantage of a category representation is that people use this categorical scheme to describe observed emotional displays in daily life. However, discrete lists of emotions fail to describe the range of emotions that occur in natural communication settings. For example, although prototypical emotions are key points of emotion reference, they cover a rather small part of our daily emotional displays.

In a video recording, facial expression and speech information are two most direct ways we can obtain. Vision-based and audio-based emotion recognition have been obtained great achievements in recent years, but still have large space to be improved. Current techniques for the detection and tracking of facial expressions are sensitive to head pose, clutter, and variations in lighting conditions, while current techniques for speech processing are sensitive to auditory noise. Audio-visual fusion can make use of the complementary information from these two channels.

Emotion recognition can be used in human computer interaction (HCI) scenarios [96], potential commercial applications of automatic human emotion recognition include systems for customer services, call centers, and intelligent automobile systems. For example, an automatic service call center with an emotion detector would be able to make an appropriate response or pass control over to human operators, while an intelligent automobile system with a fatigue detector could monitor the vigilance of the driver and apply an appropriate action to avoid accidents.

2.7 Spoken Document Retrieval (SDR)

A spoken document retrieval system allows the user to browse, search and retrieve speech information from a large database of speech signals. It presents the output ordered by relevance to some textual queries [9]. Some systems combine image retrieval, text retrieval, and video retrieval. These systems, called multimedia information retrieval (MMIR), extract semantic information from multimedia (audio, image, video etc.) data sources.

A typical SDR system has two main components. The first one is offline and consists of populating and indexing the database. Here, audio streams are first automatically or manually segmented and labeled. Then, an indexing structure is created. The second component is online. Here, the user submits a query and the system searches the indexed database and returns relevant audio segments.

Many methods for SDR have been developed in recent years. In [23], Li presented the nearest feature line (NFL) classification method for content-based audio retrieval. In this system, information is represented by multiple prototypes per class, and a nearest neighbor classifier is used. A different approach that uses the distance-from-boundary (DFB) metric for audio retrieval was proposed in [10]. In this approach, first a boundary inside the query pattern location is obtained. Then, the distance of all patterns in the database to this boundary are sorted. In [11], Kiranyaz developed a generic and robust audio-based multimedia indexing and retrieval framework that dynamically integrates audio feature extraction modules. This system also uses high-level content classification and segmentation to improve the retrieval accuracy. In [24], Kiranyaz proposed a fuzzy approach to multimedia retrieval where the input audio is segmented and classified as speech, music, fuzzy, or silent. A browsing and retrieval system was proposed in [25]. This system uses a multiple query strategy to combine audio and text and was applied to MIT spoken lecture processing. In [26], Hansen et al. presented a comprehensive spoken document retrieval system, called "SpeechFind", which includes accent classification, document expansion, speech recognition, speech segmentation, watermarking, and retrieval to address the National Gallery of the Spoken Word (NGSW) problem. A multilevel knowledge indexing and semantic verification method for SDR was proposed in [27]. This system uses three information sources: transcription data, keywords, and hypernyms of the keywords. A semantic network with forward-backward propagation is used for semantic verification of the retrieved documents. In [97], Lo developed a multi-label learning method for audio tag annotation and retrieval. This approach combines SVM and AdaBoost classifiers for tag classification, and applies probability and ranking ensembles to annotate and retrieve. In [98], Pan proposed an interaction strategy for SDR, which first retrieves results based on a short list of key terms provided by the user. This first step is modeled by a Markov decision process, and then by reinforcement learning on the related key terms.

Similar to Spoken Document Retrieval, other systems such as music information retrieval (MIR) [32], language identification and retrieval [40], use speaker/audio segmentation and feature extraction as key processing steps.

CHAPTER 3

SPEAKER SEGMENTATION

3.1 Motivations

Speaker segmentation is one of the most fundamental preprocessing steps in speech data analysis. It consists of detecting speaker changing points in the speech signal stream. Our goal is to maximize the detection of the true speaker changing points while minimizing the number of false detections.

Even though, speaker segmentation has been investigated extensively, it remains a challenging task. For instance, BIC [53] and related speaker segmentation methods [70, 71] can provide good segmentation results, but they have some limitations and challenges. First, these methods use a sliding window and are sensitive to the size of this window. A small window will not mix different speakers in the same segment. However, each segment may not have statistically sufficient samples to learn the model parameters. Conversely, a large window will have enough samples. However, this may mix different speakers in the same segment making it harder to learn model parameters that characterize each speaker. Second, a single distance metric cannot detect all changing points while keeping the false alarm rate low.

To address these problems, in this chapter, we propose speaker segmentation methods that consider multiple distance measures within the same analysis window simultaneously and fuse their results. In particular, we propose two different approaches. The first one fuses multiple extrema point sets generated by different methods. The second approach performs the fusion at the distance level and generates a single set of extrema points.

3.2 Extrema Point-level Fusion

Similar to other metric-based speaker segmentation algorithms, our approach segments an audio stream by processeing one interval window at a time. Typically, the interval window is set to 12 second. Within each analysis window, we consider different metrics to detect multiple sets of changing points. Suppose that we apply K segmentation algorithms, Seg_1, Seg_2, ..., Seg_k. Each method uses a different metric and generates one set of extrema points. The K sets are combined and similar points are merged. Two points are considered similar and merged if they are detected within 0.5 sec from each other. Our proposed fusion approach uses multiple segmentation algorithms with strict parameters. Consequently, each method detects only reliable changing points with few false alarms. This setting may cause each method to miss some true changing points. However, by combining all extrema point sets using a union operator, the number of misdetection will be minimized. The proposed extrema point-level fusion algorithm is outlined in Algorithm 3.1.

Figure 3.1 illustrates our proposed fusion method with a simple example. In this case, the speech signal contains four actual changing points marked as C, D, E, and F with vertical dash lines. We use $K = 3$ segmentation algorithms: T^2 [62], BIC [53], and $KL2$ [52]. The T^2 [62] method detected changing points at C, E, and F position. BIC [53] detected two changing points at C and D location, while $KL2$ [52] detected 5 points at A, B, D, E, and F. Thus, T^2 failed to detect point D, BIC missed points E and F, and $KL2$ missed point C, and detected 2 false alarms at A and C. As it can be seen, none of these methods detected many false alarms, and some of them missed few true changing points. The union of all changing points is {A, B, C, D, E, F}. Due to the fact that B and C are close to each other, and two algorithms detected C, point B is merged with point C. Also, since point A was detected at the very beginning of the speech signal, it can be eliminated using a heuristic constraint that any speaker changes should occur at least 1 second into the speech. Thus, the final changing points detected by our extrema point-level fusion are {C, D, E, F}.

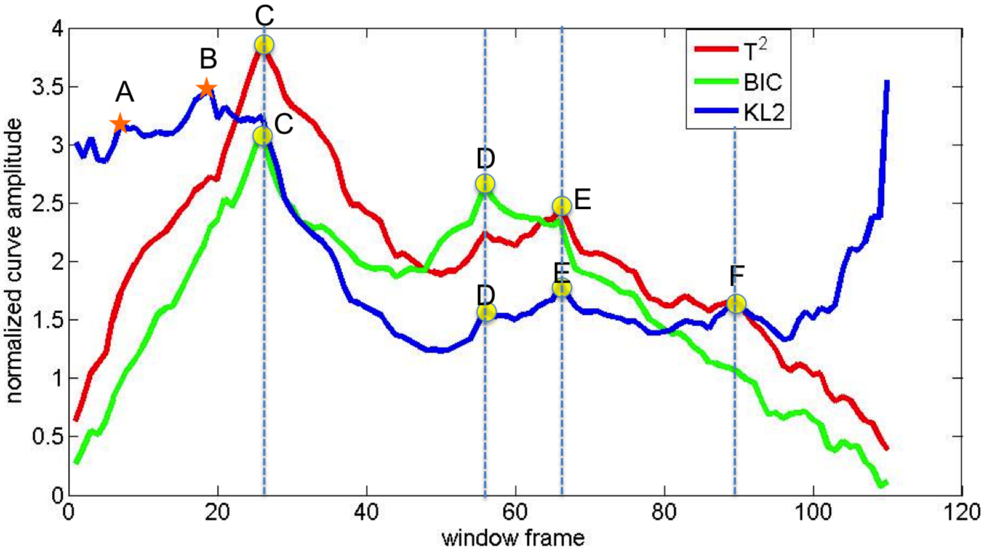


Figure 3.1: Speaker changing points detected by BIC, KL, and T^2 algorithms. Circled points are the true change points.

3.3 Distance-level Fusion

Our second segmentation approach is based on fusion of different methods at the distance level. This approach, called distance level fusion, is similar to the extrema point-level fusion in the sense that it relies on different methods to generate

Algorithm 3.1 Extrema point-level fusion

```
Initialize the interval window $W_{ini} = [a, b]$, e.g. $a = 0$, $b = 12$sec
Initialize the length of the window growing $W_g$, e.g. $W_g = 6$sec
repeat
    Detect changing points in window $[a, b]$ by different algorithms (e.g. $BIC$, $KL$, $T^2$)
    if no changes in window $[a, b]$ then
        $b = b + W_g$
    else
        for each segmentation algorithm $k$ do
            $t_{Seg_k}$: the detected changing points by $Seg_k$
        end for
        Merge all changing points detected by these $K$ methods,
            $t = \bigcup_{k=1}^{K} t_{Seg_k}$
        New starting position $a$ is set to the last changing position detected previously, $a = t_{last}$
        $b = a + length(W_{ini})$
    end if
until reach the end of the audio stream
return changing points
```

multiple hypothesis. However, instead of merging all changing points detected by each algorithm, fusion is performed at an earlier stage. First, the distance curves are normalized to have the same scale. Then, the distances are averaged to produce one simple distance curve. Finally, one set of extrema points is detected from the average distance curve. The details of the Distance level fusion algorithm is outlined in Algorithm 3.2.

Figure 3.2 illustrates the proposed distance level fusion for speech segmentation using the same signal used in Figure 3.1. In this case, the distance curves are normalized to have range values within $[0, 1]$. As it can be seen, extrema points that are consistent in multiple distance curves would also persist in the average distance curve. In this case, the fusion missed true changing point F and the two false alarms at A and B. This behavior is also observed on other longer speech segments. In other words, the distance level fusion has the ability to reduce the false alarms at the risk of missing true changing points.

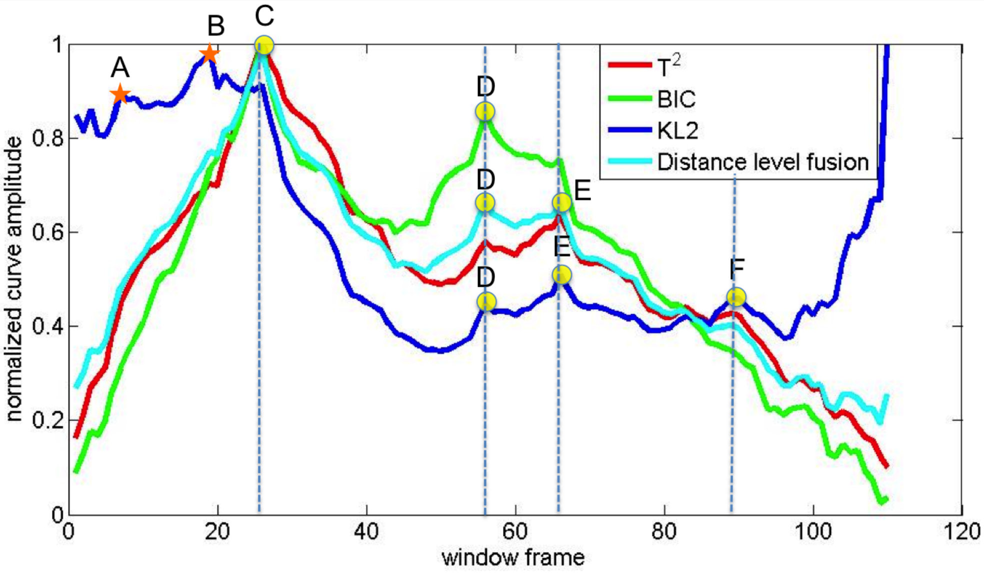


Figure 3.2: Distance curves for speaker changing points detection by BIC, KL, T^2 and the proposed distance level fusion.

Algorithm 3.2 Distance level fusion

```
Initialize the interval window $W_{ini} = [a, b]$, e.g. $a = 0$, $b = 12$sec
Use a 2-second window of MFCC audio features as the template
Initialize the sliding step of the window $W_{slide} = 5sec$
Initialize two windows: $w_1 = [a, t]$ and $w_2 = [t, b]$, $a < t < b$
repeat
    for each segmentation algorithm $k$ do
        Compute the distance curve $d_{Seg_k}(w_1, w_2)$
        Normalize the distance curve using
            $d_{Seg_k}^{Norm} = \frac{d_{Seg_k}}{\max(d_{Seg_k})}$
    end for
    Compute the fusion distance curve, $d_{fusion}$, using
        $d_{fusion}(w_1, w_2) = \sum_{k=1}^{K} a_k * d_{Seg_k}^{Norm}(w_1, w_2)$
        where $\sum a_k = 1$
    Detect local maxima points that are larger than a threshold as potential changing points
    $a = a + W_{slide}$
    $b = a + length(W_{ini})$
until reach the end of the audio stream
return changing points
```

CHAPTER 4

SPEAKER IDENTIFICATION

4.1 Motivations

A speaker identification system allows physicians to identify and retrieve speech segments of a given speaker from a large simulation video database. As shown in Fig. 4.1, a typical speaker identification system has two main components: offline training and online testing. In the offline training phase, first the audio streams are extracted from the training videos and processed by the speaker segmentation component. Second, the user assigns a class label (speaker) to each segment. Then, features are extracted from each segment. Finally, using features from all labeled segments, a classifier is trained to discriminate between segments that originated from different speakers.

In the online testing phase, the input consists of an unlabeled video recording. First, the audio component is extracted and segmented. Then, each segment is labeled by the classifier in a completely unsupervised way. As a result, the system will identify "who spoke and when".

Feature extraction is one of the most important and critical component of the speaker identification system. A good feature representation can improve the classification accuracy. Feature extraction and representation for speech data analysis is a challenging task. In fact, existing feature representation and speaker identification

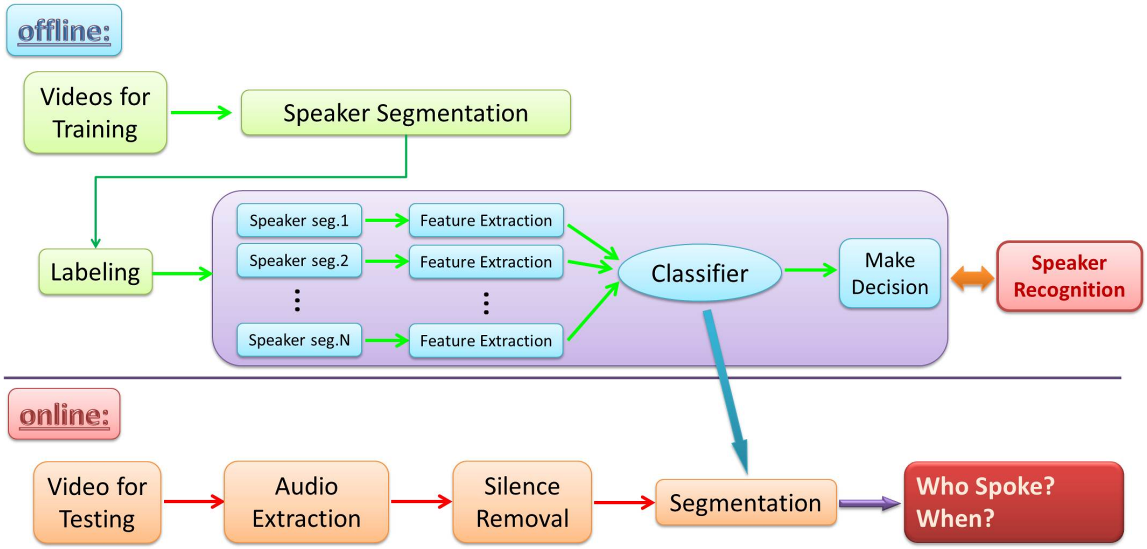


Figure 4.1: An overview of the speaker identification component of the proposed system.

algorithms [7, 39, 77, 80, 99] did not provide satisfactory performance on our considered application for the following reasons. First, different segments can have different lengths and they need to be mapped to features of equal sizes. Second, a conversation in one segment can have many interruptions. Thus, feature representation needs to be robust ignoring small segments not spoken by the main speaker. Third, speech signals tend to be too noisy when only one fixed microphone is used for all speakers.

In this chapter, we propose feature representation approaches to address these limitations. Specifically, we propose soft bag-of-word (BoW) feature representations of speech data for speaker identification. We define three types of BoW that are based on crisp, fuzzy, and possibilistic voting. Instead of working directly in the original spectral feature space, our soft BoW approach maps low-level audio features to more meaningful and interpretable histogram descriptors. Furthermore, we propose a generalization of the BoW feature representation based on an adaption of Fisher Vectors

(FV) to audio data. FV has achieved great performance in image classification [92]. It is based on the Fisher Kernel principle and it combines the benefits of generative and discriminative approaches by computing the gradient of the sample log-likelihood with respect to the model parameters.

4.2 Soft Bag-of-words Feature Representation

Inspired by the bag-of-word (BoW) feature representation methods in document classification [87] and computer vision [88], we propose a generalization of this representation that transforms low-level audio streams to more meaningful feature descriptors using two main steps: (1) vocabulary construction; and (2) membership mapping and histogram-based feature construction.

4.2.1 Visual Vocabulary Construction

We assume that we have S speakers and that for each speaker i we have a training set, $\boldsymbol{X}^i = \{\boldsymbol{x}_j^i | j = 1, ..., N^i\}$, of N^i low-level features. Each feature, $\boldsymbol{x}_j^i \in \Re^D$, is a D^{th} dimensional vector extracted from the j^{th} utterance of the i^{th} speaker.

The first step consists of summarizing each $\mathbf{X}^i$ by a set of representative prototypes $\{\boldsymbol{p}_1^i, \boldsymbol{p}_2^i, ..., \boldsymbol{p}_{K^i}^i\}$. This quantization step is achieved by partitioning $\mathbf{X}^i$ into K^i clusters and letting $\boldsymbol{p}_k^i$ be the centroid of the k^{th} partition. Any clustering algorithm can be used for this task. In our work, we use the Fuzzy C-means (FCM) [100] algorithm. The FCM partitions the N^i samples into K^i clusters by minimizing the sum of within-cluster distances, i.e.,

$$\mathbf{J}(\mathbf{U};\mathbf{X}^i) = \sum_{j=1}^{N^i}\sum_{t=1}^{K^i} \mu_{tj}^m d^2(\boldsymbol{x}_j^i, \boldsymbol{p}_t^i). \tag{4.1}$$

In (4.1), $d(\boldsymbol{x}_j^i, \boldsymbol{p}_t^i)$ refers to the Euclidean distance between feature $\boldsymbol{x}_j^i$ and prototype of cluster t, $\boldsymbol{p}_t^i$. $U = [\mu_{tj}]$ represents the membership of $\boldsymbol{x}_j^i$ in cluster t [101] and satisfies the constraints:

$$\begin{cases} \mu_{tj} \in [0,1], \\ \sum_{t=1}^{K^i} \mu_{tj} = 1 \end{cases} \tag{4.2}$$

After clustering, we obtain a set of K^i prototypes for each speaker class i. Since the prototypes of each speaker were generated independently, some of them may be similar. To reduce the computational complexity of subsequent steps, we reduce the number of prototypes by identifying similar ones and merging them. We use Hopkins statistics [102] to evaluate the distance between pairs of prototypes $\boldsymbol{p}_i$ and $\boldsymbol{p}_j$. That is, we compute

$$D(\boldsymbol{p}_i, \boldsymbol{p}_j) = \frac{\sum_{k=1}^{N} |\boldsymbol{\mu}_{ik} - \boldsymbol{\mu}_{jk}|}{\sum_{k=1}^{N} |\boldsymbol{\mu}_{ik}| + \sum_{k=1}^{N} |\boldsymbol{\mu}_{jk}|}. \tag{4.3}$$

Pairs of prototypes where $D(\boldsymbol{p}_i, \boldsymbol{p}_j)$ is less than a threshold will be merged.

Let K' be the total number of prototypes after merging similar ones. Each prototype $\boldsymbol{p}_k$ is a representative of cluster c_k that summarizes a group of similar speech segments. Let σ_k be the variance of all features $\boldsymbol{x}_j$ assigned to cluster c_k. Compared to the traditional bag-of-word approach, each cluster can be regarded as a “word”.

4.2.2 Membership Mapping and Feature Representation

Instead of using the original feature space $\boldsymbol{X}$, we map it to a new space $\boldsymbol{H}$ characterized by the K' clusters that capture the characteristics of the training data.

This mapping is defined as

$$\begin{aligned} F: &\quad \boldsymbol{x} \longrightarrow \boldsymbol{H} \\ F(\boldsymbol{x}_j) &= [f_1(\boldsymbol{x}_j), ..., f_{K'}(\boldsymbol{x}_j)] \end{aligned} \tag{4.4}$$

In (4.4), $f_i(\boldsymbol{x}_j) \in [0,1]$ is the mapping of feature $\boldsymbol{x}_j$ with respect to cluster i. This mapping can be crisp, fuzzy, or possibilistic.

4.2.3 Crisp Mapping

In crisp mapping, each feature vector x_j is assigned a binary membership value to each "word" i based on its relative distances to all words. This mapping considers only the closest word (i.e. prototype) to word i and is defined as:

$$f_i^c(\boldsymbol{x}_j) = \begin{cases} 1 & \text{if } i = \underset{k}{argmin} \parallel x_j - p_k \parallel^2 \\ 0 & otherwise \end{cases} \tag{4.5}$$

This mapping is used in the standard BoW approach [88]. It is reasonable if $\boldsymbol{x_j}$ is close to one word and far from the other words. However, if $\boldsymbol{x_j}$ is close to multiple words (i.e., $\boldsymbol{x}_j$ is located close to the clusters' boundaries), then, crisp mapping will not preserve this information.

In addition to this standard binary voting, where each sample contributes to each keyword with a binary value (1 if the keyword is the closest one to the sample and 0 otherwise), we propose generalizations that use soft voting.

4.2.4 Fuzzy Mapping

Instead of using binary voting (as in eq. (4.5)), fuzzy mapping uses soft labels to allow for partial or gradual membership values. This type of labeling offers a richer representation of belongingness and can handle uncertain cases. In particular, a sample x_j votes to each word i in the codebook with a membership degree $f_i^f(x_j)$ such that:

$$\begin{cases} f_i^f(x_j) \in [0,1] \\ \sum_{i=1}^{|K'|} f_i^f(x_j) = 1 \end{cases} \tag{4.6}$$

Many clustering algorithms use this type of labels to obtain a fuzzy partition. In the proposed fuzzy BoW (F-BoW) approach, we use the memberships derived within the Fuzzy C-Means (FCM) [100] algorithm, i.e.,

$$f_i^f(\boldsymbol{x}_j) = \frac{1}{\sum_{t=1}^{|K'|} (\frac{D_{ji}}{D_{jt}})^{\frac{2}{m-1}}}. \tag{4.7}$$

In (4.7), $m \in (1, \infty)$ is a constant that controls the degree of fuzziness, and D_{jt} is the distance between feature vector x_j and the prototype summarizing cluster t. To take into account the shape of the clusters, we use

$$D_{jt} = \sum_{k=1}^{D} \frac{||x_{jk} - p_{tk}||^2}{\sigma_{tk}^2} \tag{4.8}$$

where σ_{tk}^2 is the variance of the kth feature of cluster t and D is the dimensionality of the feature space.

4.2.5 Possibilistic Mapping

The fuzzy membership in (4.7) is a relative number that depends on the relative distance of $\boldsymbol{x_j}$ to all prototypes. It does not distinguish between samples that are equally close to multiple prototypes and samples that are equally far from all prototypes.

An alternative approach to generate soft labels is based on possibility theory [101]. Possibilistic labeling relaxes the constraint in (4.6) that the memberships across all words must sum to one. It assigns "typicality" values, $f_i^p(\boldsymbol{x}_j)$, that do not consider the *relative* position of the point to all clusters. As a result, if x_j is a noise point, then $\sum_{t=1}^{|K'|} f_t^p(\boldsymbol{x}_j) \ll 1$, and if x_j is typical of more than one cluster, we can have $\sum_{t=1}^{|K'|} f_t^p(\boldsymbol{x}_j) > 1$. Many robust partitional clustering algorithms [103, 104] use this type of labeling in each iteration. In this paper, we use the membership function derived within the Possibilistic C-Means [101], i.e.,

$$f_i^p(\boldsymbol{x}_j) = \frac{1}{1 + (\frac{D_{ji}}{\eta_j})^{\frac{2}{m-1}}}. \tag{4.9}$$

In (4.9), η_j is a cluster-dependent resolution/scale parameter [101], $m \in (1, \infty)$, and D_{ji} is as defined in (4.8).

Robust statistical estimators, such as M-estimators and W-estimators [105], use this type of memberships to reduce the effect of noise and outliers.

4.3 Fisher Vector Feature Representation

Fisher Vector (FV) was proposed in [92] for fine-grained image data and is based on the Fisher Kernel principle [79]. Fisher kernel combines the benefits of generative and discriminative approaches by computing the gradient of the sample log-likelihood with respect to the model parameters.

FV feature representation is a generalization of the bag-of-words approach and has achieved great performance in image classification [92–95].

4.3.1 Fisher Kernel and Fisher Vector

Let $Z = [z_1, z_2, ..., z_T]$ be a sample of T observations and let u_λ be a probability density function with parameters $\lambda = [\lambda_1, ..., \lambda_M]$ that models the generative process of the elements of Z. The score function can be represented by the gradient of the log-likelihood of the model [92] as:

$$G_\lambda^Z = \bigtriangledown_\lambda \log u_\lambda(Z) \tag{4.10}$$

G_λ^Z indicates how the parameters of the generative model u_λ should be modified to better fit the data Z.

The Fisher Kernel (FK) [79] uses the score function to define the similarity between two samples Z and P as:

$$\begin{aligned} K_{FK}(Z, P) &= G_\lambda^{Z\prime} F_\lambda^{-1} G_\lambda^P \\ &= g_\lambda^{Z\prime} g_\lambda^P \end{aligned} \tag{4.11}$$

where F_λ denotes the Fisher Information Matrix (FIM) [92] and is defined as:

$$F_\lambda = E_{x \sim u_\lambda}[G_\lambda^x G_\lambda^{x\prime}] \tag{4.12}$$

In (4.11), g_λ^Z denotes the normalized gradient vector, also called the Fisher Vector (FV) [92], of sample Z. Using Cholesky decomposition, $F_\lambda^{-1} = L_\lambda{}' L_\lambda$, the FV can be represent as:

$$g_\lambda^Z = L_\lambda G_\lambda^Z = L_\lambda \bigtriangledown_\lambda \log u_\lambda(Z). \tag{4.13}$$

In the following, we adapt the FV feature representation to the problem of speaker identification. We use it to map maps low-level audio features from multiple small segments to a high dimensional vector.

4.3.2 Fisher Vector Features for Speech Data

Assume that we have a training set $\boldsymbol{X} = \{\boldsymbol{X}_1, ..., \boldsymbol{X}_t, ..., \boldsymbol{X}_T\}$ of T speech segments generated from S speakers. Each segment t is decomposed into N^t small overlapping window frames and a low-level feature $\boldsymbol{x}_t^i$ (e.g. MFCC or PLP) is extracted from each frame. Thus, $\boldsymbol{X}_t = \{\boldsymbol{x}_t^1, ..., \boldsymbol{x}_t^i ..., \boldsymbol{x}_t^{N^t}\}$ where $\boldsymbol{x}_t^i$ is a D dimensional feature vector.

Assuming that speech segments are independent, using (4.13), the FV for segment $\boldsymbol{X}_t$ can be represented as:

$$g_\lambda^{\boldsymbol{X}_t} = \sum_{i=1}^{N^t} L_\lambda \bigtriangledown_\lambda \log u_\lambda(\boldsymbol{x}_t^i) \tag{4.14}$$

In the following, we assume that u_λ is modeled by a mixture of K Gaussian components with parameters $\lambda = \{w_k, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k, k = 1, ..., K\}$, where w_k, $\boldsymbol{\mu}_k$ and $\boldsymbol{\Sigma}_k$ denote the mixture weights, mean vector, and covariance matrix of Gaussian k respectively, and $w_k \geq 0$, $\sum_{k=1}^K w_k = 1$. That is,

$$u_\lambda(\boldsymbol{x}_t^i) = \sum_{k=1}^{K} w_k u_k(\boldsymbol{x}_t^i) \tag{4.15}$$

where $u_k(\boldsymbol{x}_t^i)$ is the Gaussian function:

$$u_k(\boldsymbol{x}_t^i) = \frac{1}{(2\pi)^{D/2}|\boldsymbol{\Sigma}_k|^{1/2}} exp\{-\frac{1}{2}(\boldsymbol{x}_t^i - \boldsymbol{\mu}_k)' {\boldsymbol{\Sigma}_k}^{-1}(\boldsymbol{x}_t^i - \boldsymbol{\mu}_k)\} \tag{4.16}$$

As in [92], we use the soft-max formalism [106] to ensure that the weights are positive, that is we let:

$$w_k = \frac{\exp(\alpha_k)}{\sum_{j=1}^K \exp(\alpha_j)} \tag{4.17}$$

It can be shown [92] that the gradients of the parameters of the GMM are given by:

$$\bigtriangledown_{\alpha_k} \log u_\lambda(\boldsymbol{x}_t^i) = \gamma_t^i(k) - w_k, \tag{4.18}$$

$$\bigtriangledown_{\mu_k} \log u_\lambda(\boldsymbol{x}_t^i) = \gamma_t^i(k)(\frac{\boldsymbol{x}_t^i - \boldsymbol{\mu}_k}{\boldsymbol{\sigma}_k^2}), \tag{4.19}$$

and

$$\bigtriangledown_{\boldsymbol{\sigma}_k} \log u_\lambda(\boldsymbol{x}_t^i) = \gamma_t^i(k)[\frac{(\boldsymbol{x}_t^i - \boldsymbol{\mu}_k)^2}{\boldsymbol{\sigma}_k^3} - \frac{1}{\boldsymbol{\sigma}_k}]. \tag{4.20}$$

In (4.18) - (4.20), $\gamma_t^i(k)$ is the posterior probability of assigning feature vector $\boldsymbol{x}_t^i$ to the k-th Gaussian component and is given by

$$\gamma_t^i(k) = \frac{w_k u_k(\boldsymbol{x}_t^i)}{\sum_{m=1}^{K} w_m u_m(\boldsymbol{x}_t^i)} \tag{4.21}$$

The parameter L_λ for the FV representation in (4.13) can be represent as the square-root of the inverse of the FIM [92]. Thus, the normalized gradients can be represented as:

$$g_{\alpha_k}^{\boldsymbol{X}_t} = \frac{1}{\sqrt{w_k}} \sum_{i=1}^{N^t} (\gamma_t^i(k) - w_k), \tag{4.22}$$

$$g_{\mu_k}^{\boldsymbol{X}_t} = \frac{1}{\sqrt{w_k}} \sum_{i=1}^{N^t} \gamma_t^i(k)(\frac{\boldsymbol{x}_t^i - \boldsymbol{\mu}_k}{\boldsymbol{\sigma}_k}), \tag{4.23}$$

and

$$g_{\sigma_k}^{\boldsymbol{X}_t} = \frac{1}{\sqrt{w_k}} \sum_{i=1}^{N^t} \gamma_t^i(k) \frac{1}{\sqrt{2}} [\frac{(\boldsymbol{x}_t^i - \boldsymbol{\mu}_k)^2}{\boldsymbol{\sigma}_k^2} - 1] \tag{4.24}$$

The final FV feature representation for a speech segment, $\boldsymbol{X}_t$ is defined as the concatenation of the normalized gradients in (4.22) - (4.24) of all K components. That is,

$$g_\lambda^{\boldsymbol{X}_t} = [g_{\alpha_1}^{\boldsymbol{X}_t}, ..., g_{\alpha_K}^{\boldsymbol{X}_t}, g_{\boldsymbol{\mu}_1}^{\boldsymbol{X}_t}, ..., g_{\boldsymbol{\mu}_K}^{\boldsymbol{X}_t}, g_{\boldsymbol{\sigma}_1}^{\boldsymbol{X}_t}, ..., g_{\boldsymbol{\sigma}_K}^{\boldsymbol{X}_t}] \tag{4.25}$$

In (4.22) - (4.24), $g_{\alpha_k}^{\boldsymbol{X}_t}$ is a scalar, $g_{\mu_k}^{\boldsymbol{X}_t}$ and $g_{\sigma_k}^{\boldsymbol{X}_t}$ are D dimensional vectors. Thus, the dimension of the FV in (4.25) is $(2D+1)K$. One key advantage of the FV feature representation is that each speech segment is mapped to a $(2D+1)K$ dimensional vector regardless of the duration of the segment. This is a desirable feature since speech segmentation algorithms generate segments with variable size.

4.4 Classification Algorithms for Speaker Identification

After feature mapping, each segment needs to be labeled using a classifier. In the following, we outline classifiers that proved to be effective with our proposed BoW and FV feature representations.

4.4.1 K-NN classifier

K-NN classifiers are appealing because of their simplicity, ability to model non-parametric distributions, and theoretical optimality as the size of the training data goes to infinity. A common drawback of the standard or crisp K-NN classification rule [107] is that the K nearest training patterns are treated equally important in the confidence assignment of the test pattern. This may degrade the classifier's accuracy in regions where patterns from different classes overlap. To overcome this limitation, we use the fuzzy K-NN [108] where the confidence value assigned to pattern $\boldsymbol{x}$ in class i is computed using

$$\mu^i(\boldsymbol{x}) = \sum_{k=1}^{K} \widetilde{\mu}^i(\boldsymbol{y}_k)\omega(\boldsymbol{x}, \boldsymbol{y}_k). \tag{4.26}$$

In (4.26), $\widetilde{\mu}^i(\boldsymbol{y}_k)$ is a fuzzy membership assigned to each training sample $\boldsymbol{y}_k$ in class i. These memberships are assigned using

$$\widetilde{\mu}^i(\boldsymbol{y}_k) = \begin{cases} 0.51 + (\frac{n_i}{K}) \times 0.49, \text{ if } i = j \\ \\ (\frac{n_i}{K}) \times 0.49, \text{ if } i \neq j \end{cases} \tag{4.27}$$

where n_i denotes the number of neighbors of $\boldsymbol{y}_k$ that belong to the i^{th} class, i.e., $\sum_{i=1}^{C} n_i = K$, and j is the actual class label of sample $\boldsymbol{y}_k$.

In (4.26),

$$\omega(x, y_k) = \frac{(1/\|\boldsymbol{x} - \boldsymbol{y}_k\|^{2/(m-1)})}{\sum_{k=1}^{K}(1/\|\boldsymbol{x} - \boldsymbol{y}_k\|^{2/(m-1)})} \tag{4.28}$$

In other words, the confidence value assigned to a test pattern depends on the membership degrees of the K-NNs and their relative proximity.

4.4.2 Naive Bayes Classifier

Assume that we have a set of labeled speech segments $X = \{X^i\}$, C classes $[S_1, ..., S_j, ..., S_C]$, and representative vocabularies (i.e. codebook or cluster centers) $V = \{v_t\}$. Let $f_t(X^i)$ denotes the value in bin v_t of the histogram representing segment X^i. To classify a new test sample, X^s, Bayes' rule is applied and the maximum a posteriori score is used for prediction. That is,

$$P(S_j|X^s) \propto P(S_j)P(X^s|S_j) = P(S_j)\prod_{t=1}^{|V|} P(v_t|S_j)^{f_t(X^s)} \tag{4.29}$$

In (4.29), $P(S_j)$ is the a priori probability of class S_j, and the class-conditional probability $P(v_t|S_j)$ denotes the probability of word v_t occurring in class S_j and can be estimated using:

$$P(v_t|S_j) = \frac{\sum_{X^i \in S_j} f_t(X^i)}{\sum_{n=1}^{|V|}\sum_{X^i \in S_j} f_n(X^i)} \tag{4.30}$$

In order to avoid the zero probability estimation in (4.30), the Laplace smoothing is frequently used, and (4.30) can be replaced with:

$$P_{Lap}(v_t|S_j) = \frac{1 + \sum_{X^i \in S_j} f_t(X^i)}{|V| + \sum_{n=1}^{|V|}\sum_{X^i \in S_j} f_n(X^i)} \tag{4.31}$$

4.4.3 Support Vector Machines (SVM) Classifier

The objective of the Support Vector Machine (SVM) classifier [109] is to find the optimal hyperplane that is a function of the features (predictive variables), such that samples on one side of the hyperplane are positive and negative on the other side. SVM classifier is more efficient than other classifiers, in terms of system performance, convergence during training and also the ability to give more accurate and generalizable classifiers. In addition to performing linear classification, SVMs can also efficiently perform a non-linear classification using the kernel trick, implicitly mapping their inputs into high-dimensional feature spaces. It has been applied to various classification tasks [37, 66, 80, 82, 92, 110–112].

The SVM classifier was initially designed for binary classification problems. It has also been extended to multi-class classification [113]. Several methods construct a multi-class classifier by combining several binary classifiers, e.g. ”one-against-all”, ”one-against-one”, and DAGSVM [114]. In this thesis, due to the limited size of training data for some speaker classes, we use ”one-against-all” SVM with linear kernel to perform multi-class speaker identification.

CHAPTER 5

SEMI-SUPERVISED SPEAKER IDENTIFICATION

5.1 Motivations

In supervised learning applications, such as speaker identification, all the training data need to be labeled. However, labeling speech data is a tedious and time consuming task. It requires human segmentation and annotation, and may not be practical for large scale datasets. In contrast, unlabeled speech data can be easily generated in large quantity and can provide useful information.

Semi-supervised learning [115] is a class of algorithms that uses both labeled and unlabeled data for learning. Typically, most of the data is unlabeled and only a small subset is labeled and used to guide the learning process. Several semi-supervised learning algorithms have been developed. Examples includes label propagation [116], local and global consistency [117], graph kernels by spectral transforms [118], and Gaussian field and Harmonic function [119].

In our proposed speaker segmentation and identification system, speech segments are automatically generated by the speaker segmentation component. Thus, a large number of speech segments can be generated. Labeling each speech segment is time consuming and may not be reliable as some segments can contain multiple speakers (due to inaccurate segmentation). As an alternative to supervised learning, we propose using a semi-supervised approach to build the speaker identification component of our system.

For each speaker present in the simulation session, we select only few segments and label them. The remaining segments are used without labels. First, we extract the Fisher Vector features as described in Chapter 4 to all labeled and unlabeled segments. Then, we use the label propagation approach [116] to learn the speakers' identity for the unlabeled speech segments as well as new test segments.

5.2 Label Propagation

Assume that we have a training set $\mathcal{X} = \{\boldsymbol{X}_1, ..., \boldsymbol{X}_l, \boldsymbol{X}_{l+1}, ..., \boldsymbol{X}_{l+u}\}$ of $l + u$ samples generated from S classes. Each $\boldsymbol{X}_i$ is a D dimensional feature vector. Within the training set $\mathcal{X}$, we have l labeled samples $\mathcal{X}_{\mathcal{L}} = \{\boldsymbol{X}_1, ..., \boldsymbol{X}_l\}$ with labels $\mathcal{Y}_{\mathcal{L}}$ $= \{y_1, ..., y_l\}$. The remaining samples $\mathcal{X}_{\mathcal{U}} = \{\boldsymbol{X}_{l+1}, ..., \boldsymbol{X}_{l+u}\}$ are unlabeled. That is their labels $\mathcal{Y}_{\mathcal{U}} = \{y_{l+1}, ..., y_{l+u}\}$ are unobserved. Typically, only a small set of the training data is labeled, that is $l \ll u$.

The objective in semi-supervised learning is to learn the labels $\mathcal{Y}_{\mathcal{U}}$ from $\mathcal{X}$ and $\mathcal{Y}_{\mathcal{L}}$. This objective can be achieved using the label propagation algorithm [116]. This algorithm is based on the assumption that data points that are close to each other tend to have similar class labels.

5.2.1 Label Propagation Algorithm

A fully connected graph is created from the whole training set $\mathcal{X}$. Each data point is represented by a node in the graph. The weight connecting node i and j, w_{ij}, reflects the similarity between the nodes and is computed using

$$w_{ij} = \exp(-\frac{d_{ij}^2}{\sigma^2}) \tag{5.1}$$

In (5.1), d_{ij}^2 is a distance measure, usually Euclidean distance, between feature vectors $\boldsymbol{X}_i$ and $\boldsymbol{X}_j$, and σ is a parameter that controls the rate of the decay.

The probabilistic transition matrix, T, is defined as:

$$T_{ij} = P(j \rightarrow i) = \frac{w_{ij}}{\sum_{k=1}^{l+u} w_{kj}}. \tag{5.2}$$

Here, T is a $(l + u) \times (l + u)$ matrix, where element T_{ij} denotes the probability to jump from node j to node i. T is normalized using:

$$\bar{T}_{ij} = \frac{T_{ij}}{\sum_k T_{ik}} \tag{5.3}$$

The $(l + u) \times S$ label matrix, Y, is defined as:

$$Y = \begin{bmatrix} Y_L \\ Y_U \end{bmatrix} \tag{5.4}$$

For the labeled subset Y_L of the data, $Y_{ij} = 1$ if the class of $\boldsymbol{X}_i$ is S_j and $Y_{ij} = 0$ otherwise. For the unlabeled subset, Y_U, the labels are initialized to arbitrary values.

The label propagation algorithm updates the label matrix Y using

$$Y \leftarrow \bar{T}Y \tag{5.5}$$

Using the split of $\bar{T}$:

$$\bar{T} = \begin{bmatrix} \bar{T}_{ll} & \bar{T}_{lu} \\ \bar{T}_{ul} & \bar{T}_{uu} \end{bmatrix} \tag{5.6}$$

The label propagation of the unlabeled component of Y is:

$$Y_U \leftarrow \bar{T}_{uu}Y_U + \bar{T}_{ul}Y_L \tag{5.7}$$

It can be shown [116] that the labels of the unlabeled data can be calculated using

$$Y_U = (I - \bar{T}_{uu})^{-1} \bar{T}_{ul} Y_L \tag{5.8}$$

As it can be seen from (5.8), given a training dataset with partially labeled samples, labels for the unlabeled subset can be learned in a direct and non-iterative way. In the following, we will apply this label propagation algorithm to speech segments for the purpose of developing a speaker identification algorithm.

5.3 Speaker Identification with Label Propagation

Assume that we have a training set $\mathcal{X} = \{\boldsymbol{X}_1, ..., \boldsymbol{X}_t, ..., \boldsymbol{X}_T\}$ of T speech segments generated from S speakers. Each segment t is decomposed into N^t small overlapping window frames and a low-level feature $\boldsymbol{x}_t^i$ (e.g. MFCC or PLP) is extracted from each frame. Thus, $\boldsymbol{X}_t = \{\boldsymbol{x}_t^1, ..., \boldsymbol{x}_t^i ..., \boldsymbol{x}_t^{N^t}\}$ where $\boldsymbol{x}_t^i$ is a D dimensional feature vector. Each speech segment can be represented by a feature matrix with different number of columns.

In chapter 4, we proposed our soft bag-of-words feature representation and the Fisher Vector (FV) representation methods. Both approaches map each speech segment to a fixed dimensional vector regardless of the duration of the segment. This is a desirable feature since speech segmentation algorithms generate segments with variable size, and most learning algorithms require features of equal dimensions.

Given that only a small group of speech segments can be labeled within a reasonable amount of time, we use the label propagation algorithm to generate labels for the remaining speech segments. Our proposed semi-supervised speaker identification with label propagation using the Fisher Vector representation has two main steps.

First, FV features, as described in section 4.3.2, are constructed. Second, we apply the label propagation algorithm to label the remaining data samples. The details of the algorithm is outlined in Algorithm 5.1.

Algorithm 5.1 Fisher Vector-based Speaker Identification using Semi-supervised Learning with Label Propagation

```
Given speech segments $\mathcal{X} = \{\boldsymbol{X}_1, ..., \boldsymbol{X}_t, ..., \boldsymbol{X}_T\}$
Assume that a small subset of $t$ segments are labeled. The remaining $T-t$ samples are unlabeled
Extract the low-level features (e.g. MFCC, PLP) for each $\boldsymbol{X}_i$
Initialize the number of Gaussian components, $K$, e.g. $K = 100$
Using all training features and (4.15), estimate the Gaussian parameters
for each speech segment $\boldsymbol{X}_t$ do
    Compute the FV feature representation, $g_\lambda^{\boldsymbol{X}_t}$ in (4.25), based on equations (4.14)-(4.24)
end for
Compute the pairwise similarity matrix $W$ using equation (5.1)
Compute the probabilistic transition matrix $T$ using equation (5.2)
Normalize $T$ using (5.3)
Compute the $Y$ matrix in equation (5.4)
Estimate the labels of the unlabeled samples, $Y_U$, using equation (5.8)
return $Y_U$
```

CHAPTER 6

EXPERIMENTAL RESULTS AND ANALYSIS

6.1 Data Collections

We use multiple data sets to validate our proposed algorithms, learn their optimal parameters, and compare them to existing algorithms. In particular, we use recordings of 15 medical simulations. All videos contain 4 speakers, most of them are female speakers. Three videos have low quality due to background noise and frequent low pitch speech. Three other videos have good quality, where speech is clear. The remaining 9 videos have fair quality with some noise and interruptions. Table 6.1 summarizes the characteristics of these video collections.

For training and evaluation purposes, each video is manually segmented and analyzed to extract the ground truth by identifying the speaker change points and labeling each segment according to the speaker. This process is tedious and may have an up to 0.5s error tolerance, which can be ignored for the purpose of our experiments.

6.2 Data Preprocessing

The medical simulations used for our experiments are available in video format. Currently, we only use the audio information to perform speaker segmentation and recognition. This is because the video resolution is low. Moreover, most conversation information is contained in the audio stream.

TABLE 6.1

Data collections used to analyze and evaluate the various speaker segmentation/recognition algorithms.

Videos	Lengths	# of Speakers	# Male	# Female	Audio Quality
Med1	6m35s	4	0	4	Fair
Med2	7m13s	4	1	3	Fair
Med3	10m20s	4	0	4	Fair
Med4	18m02s	4	1	3	Good
Med5	9m40s	4	0	4	Good
Med6	7m22s	4	0	4	Fair
Med7	10m16s	4	0	4	Fair
Med8	4m33s	4	0	4	Low
Med9	6m54s	4	1	3	Good
Med10	5m32s	4	1	3	Fair
Med11	6m43s	4	0	4	Low
Med12	7m45s	4	0	4	Fair
Med13	12m1s	4	1	3	Fair
Med14	5m34s	4	1	3	Fair
Med15	7m55s	4	0	4	Low

The recording of most of these simulations is very noisy. First, only one microphone, placed in the middle of the room, is used. Additional noise can be attributed to the frequent opening and closing of the door, walking around, and echo in the room. Another noise source is caused by the electromagnetic interference in the microphone instrument. Thus, speech enhancement is needed. In all of our experiments, we applied the spectral subtraction method [120] to reduce the noise and enhance the speech information.

In the following, we outline our approach to preprocess the data and extract useful features to achieve our objectives.

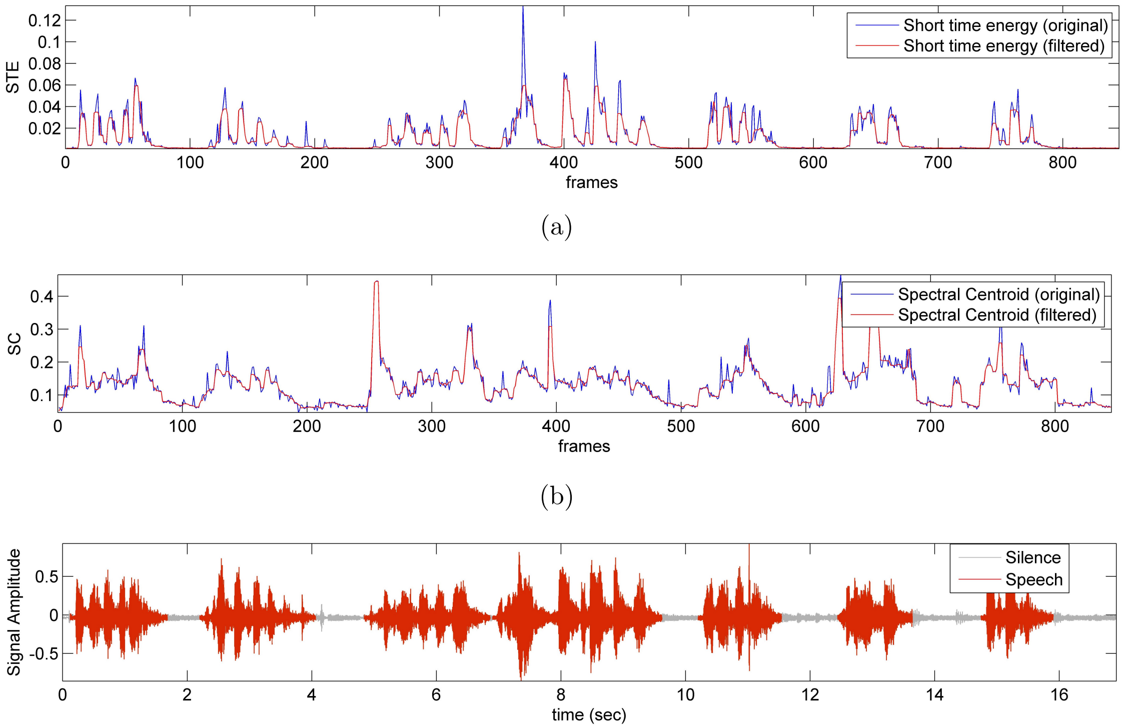
6.3 Detection and Removal of Silence Segments

The recorded audio streams include many silence segments. These segments can occur at the speaker change points and even within the same speaker's segment. They could affect the performance of speaker segmentation and other subsequent steps. Therefore, it is necessary to detect and remove as many silence segments as possible. We have implemented and compared two silence detection approaches. The first one is threshold-based while the second one is classification-based.

6.3.1 Threshold-based Silence Detection

As mentioned earlier (section 2.2.2), short-time energy (STE) and spectral centroid (SC) features are two of the most effective features in discriminating between speech and silence [47]. Thus, we use these features in our proposed system. First, we extract the STE feature (equation(2.9)) and the SC feature(equation(2.13)). Then, we compute a threshold value for these features (using equation(2.14), where f^i refers to either the STE or SC feature). Audio segments where both the STE and SC features are below the threshold are identified and considered as silence.

Fig. 6.1 uses a 17min audio stream to illustrate the threshold-based silence detection. Fig. 6.1(a) shows the STE feature (in blue) computed using equation(2.9) with a window size (variable x in equation(2.9)) of 20ms. The x-axis corresponds to the frame number (50 frames in one second), and the y-axis denotes the STE value. Typically, a moving average smoothing filter is applied to the STE to reduce noise and variability. The red plot in Fig. 6.1(a) displays the filtered STE. Fig. 6.1(b) shows the SC (in blue) computed using equation(2.13) with a window size (variable x in equation(2.13)) of 20ms. The x-axis corresponds to the frame number, and the



(a)

(b)

(c)

Figure 6.1: Threshold-based Silence Detection. (a) original STE values (in blue) and filtered STE values (in red) by a moving average smoothing; (b) original SC values (in blue) and filtered SC values (in red) by a moving average smoothing; (c) silence detection results, red represents speech sections and gray denotes detected silence.

y-axis denotes the SC value. Also, a moving average smoothing filter is applied to the SC to reduce noise and variability. The red plot in Fig. 6.1(b) displays the filtered SC. Based on the thresholds of STE (=0.003) and SC (=0.1), the silence detection results are shown in Fig. 6.1(c), where the audio signal in red is the detected speech component, while the gray color denotes the detected silence segments. A comparison of the segmented results in Fig. 6.1(c) to the ground truth shows that, for this example, the STE and SC features were able to correctly detect the silence segments.

The STE and SC approaches are quite simple and efficient. In general, they can detect silence correctly when the audio stream contains little noise (or clear background) and the speakers speak with little linking. If this is not the case, the threshold would be affected by the noisy background, and the algorithms would misclassify some speech as silence. An example of this scenario is illustrated in Fig. 6.2. Fig. 6.2(a) and 6.2(b) display the original and filtered STE and SC features of a noisy audio stream. Fig. 6.2(c) displays the results where several speech segments are misclassified as silence. In this figure, segments 1 and 3 are correctly classified as silence. However, segments 2, 4, 5, and 6 are speech segments misclassified as silence. For instance, segment 4 combines both speech and silence and fails to isolate silence. Other situations that may lead to misclassification of speech as silence may include audio segments where the whole utterance is too short, e.g. less than 2 seconds.

6.3.2 Classification-based Silence Detection

Another silence detection approach that we use in our proposed system is based on pattern classification. In particular, we use a trainable support vector machine (SVM) classifier [121]. For this approach, a labeled collection of silence utterances and non-silence or speech utterances is used to train the classifier. For features, we extract STE, ZCR and SC from each utterance. Even though many other features could be used within this approach (for example, Pitch [122], Energy Entropy [123]), our preliminary experiments have indicated that those three features are sufficient for silence detection and more importantly are efficient to compute. Fig. 6.3 displays a scatter plot of the STE, ZCR and SC features for a set of speech and silence segments. As it can be seen, these features can provide good separation between the two classes.

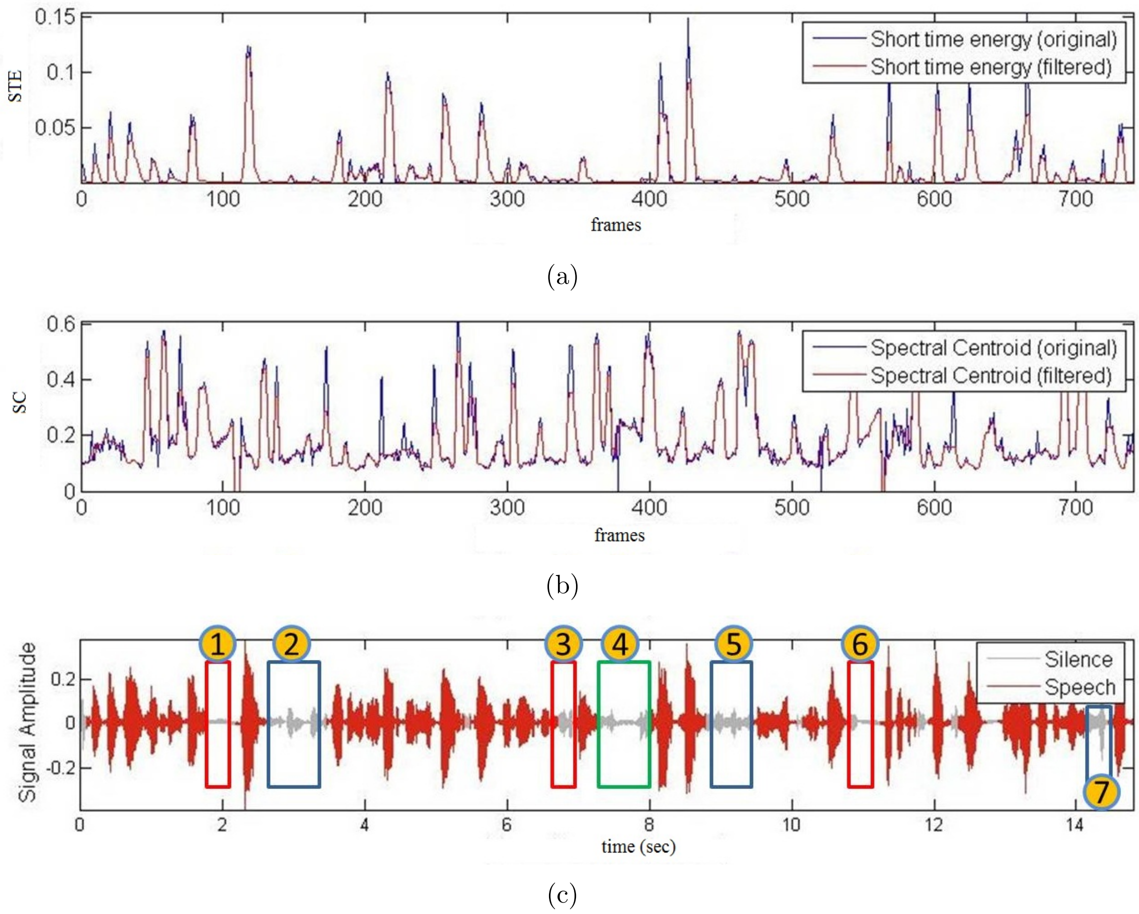


Figure 6.2: An example where threshold-based silence detection methods perform poorly.

Fig. 6.4 shows the detected silence segments using the classification-based method for the same audio signal used in Fig. 6.1. For this noise-free audio segment, the classification-based method, like the threshold-based method, detects all silence segments correctly.

Fig. 6.5 and Fig. 6.6 provide detailed comparisons of the classification-based method and the threshold-based method for two noisy audio streams. In Fig. 6.5(a) and Fig. 6.6(a), blue boxes indicate speech segments that were misclassified as silence and red boxes indicate correctly classified silence segments by the threshold based

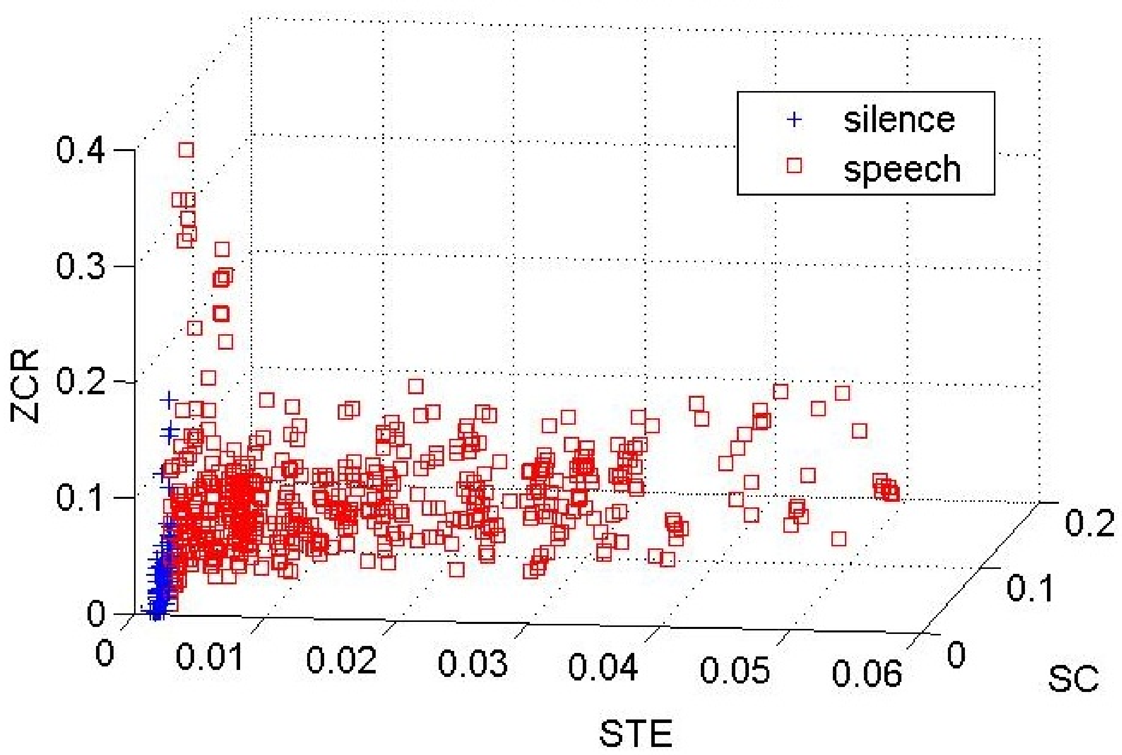


Figure 6.3: A scatter plot of the three features used to discriminate between silence and speech audio segments.

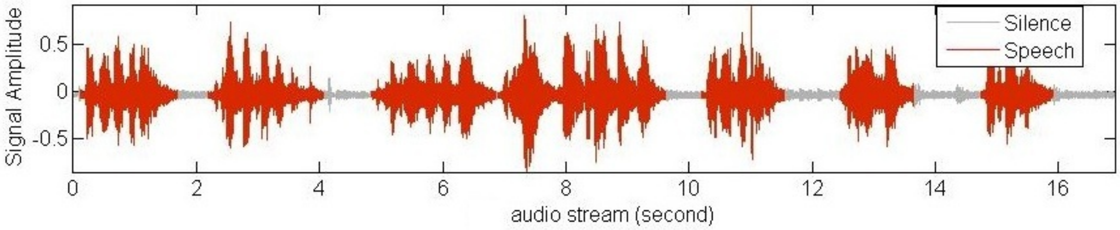
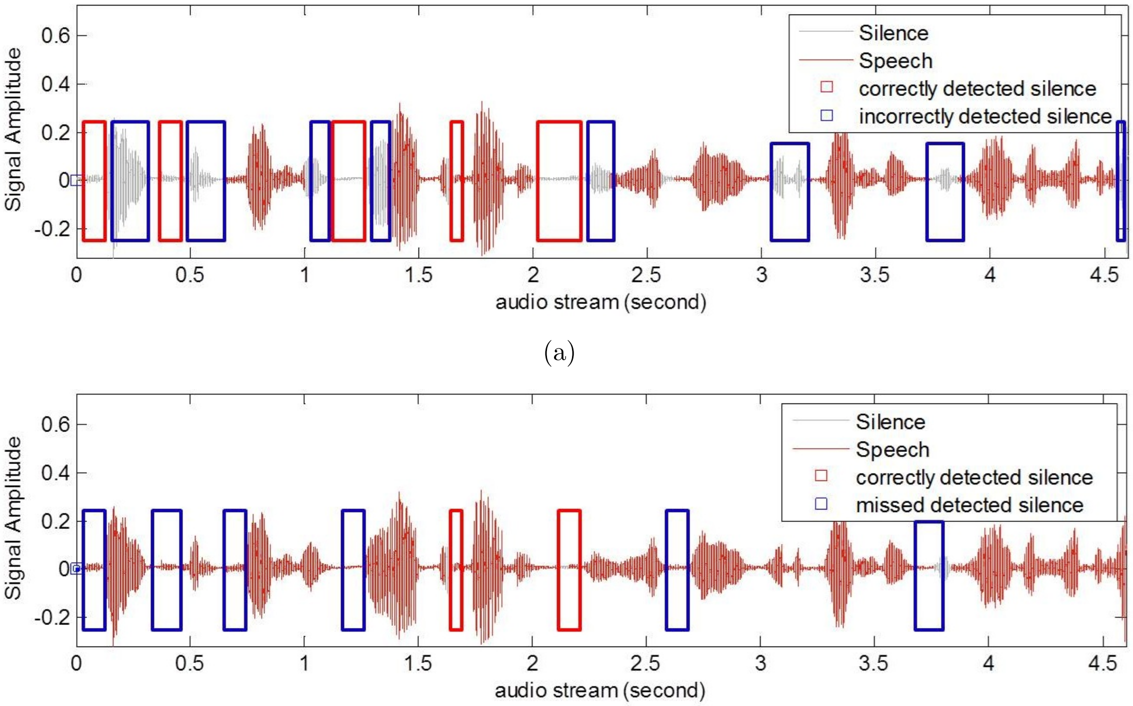


Figure 6.4: Silence detection results based on SVM classification for the audio stream used in Fig. 6.1.

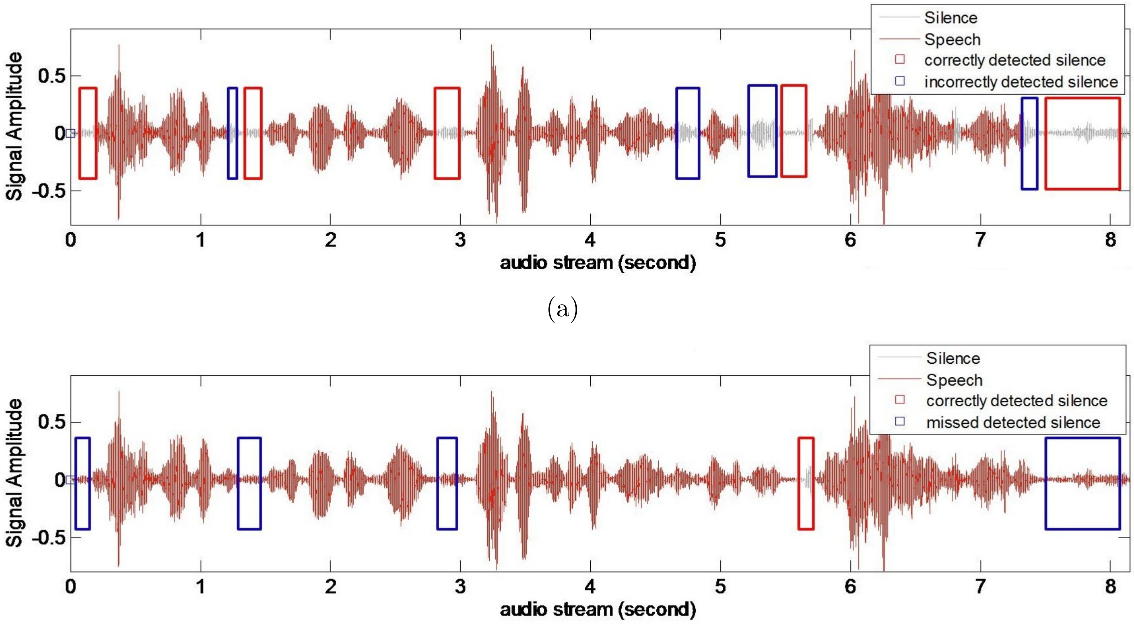


(a)

(b)

Figure 6.5: Silence segments detected using (a) threshold-based method and (b) classification-based method for noisy audio stream 1.

method. In Fig. 6.5(b) and Fig. 6.6(b), blue boxes indicate speech segments that were misclassified as speech and red boxes indicate correctly classified silence segments by the classification based method. First, we note that the threshold-based method can detect most silence segments in the audio stream. Second, the classification-based approach generates less false positives than the threshold-based approach. In our application, the cost of misclassifying speech as silence is much higher than the cost of not detecting silence. The reason is that undetected silence segments will be processed by subsequent steps and their labels could change. On the other hand, speech segments misclassified as silence will be deleted and the system cannot recover from those errors. Therefore, in our experiments, we use the classification-based approach to preprocess the data for speaker identification and emotion recognition.



(a)

(b)

Figure 6.6: Silence segments detected using (a) threshold-based method and (b) classification-based method for noisy audio stream 2.

6.4 Feature Extraction and Mapping

6.4.1 Low-level Features

In our experiments, we use several low-level features, including Mel-Frequency Cepstral Coefficient (MFCC) [36], Perceptual Linear Prediction (PLP) [41], linear prediction cepstral coefficients (LPCC) [124], Gabor Filtering Cepstral Coefficient (GFCC) [125], as well as the delta variations of these features [126], for speaker segmentation, speaker identification, and emotion recognition tasks. For GFCC, instead of using tensor decomposition as proposed in [125], we simply average all Gabor filtered spectrum features along the scales and phases to reduce the computational complexity. In particular, for a speech segment of arbitrary length, the signal is represented as a one channel or two channels digital waveform with amplitude and sampling frequency. To extract the low-level features, as described in section 2.2.1, the signal is first decomposed into small frames using a 25ms analysis window with 10ms overlap. Then, for each window, 12-dimension MFCC, PLP, LPCC, and GFCC features are extracted. Audio segments of different size would results in different number of low-level features. The flowchart of the feature extraction process is illustrated in Fig. 6.7.

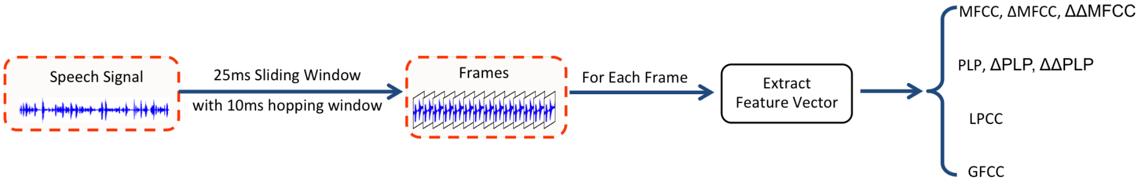


Figure 6.7: Main steps involved in extracting low-level audio features.

6.4.2 Soft Bag-of-Words Feature Mapping

All low-level features extracted from one segment are mapped to a histogram using our proposed BoW feature mapping, as described in section 4.2. Let W^i be the number of windows within a given segment i, and let $\mathbf{x}_j$ be the low-level feature (MFCC, PLP, LPCC, or GFCC) of each window j. First, we cluster the training data using the Fuzzy C-means clustering algorithm [100] to find the optimal set of prototypes. The initial number of clusters is set to K. After combining clusters from all classes and merging similar ones, we obtain a total of K' clusters. Then, each feature $\mathbf{x}_j$ is mapped to a histogram h_j with K' bins using crisp mapping (4.5), fuzzy mapping (4.7), or possibilistic mapping (4.9). Finally, we compute the normalized histogram representing segment i using

$$\overline{\boldsymbol{H}}_{\boldsymbol{i}} = \frac{\sum_{j=1}^{W^i} \boldsymbol{h}_j}{max_{k \in K'}(\sum_{j=1}^{W^i} h_{jk})}. \tag{6.1}$$

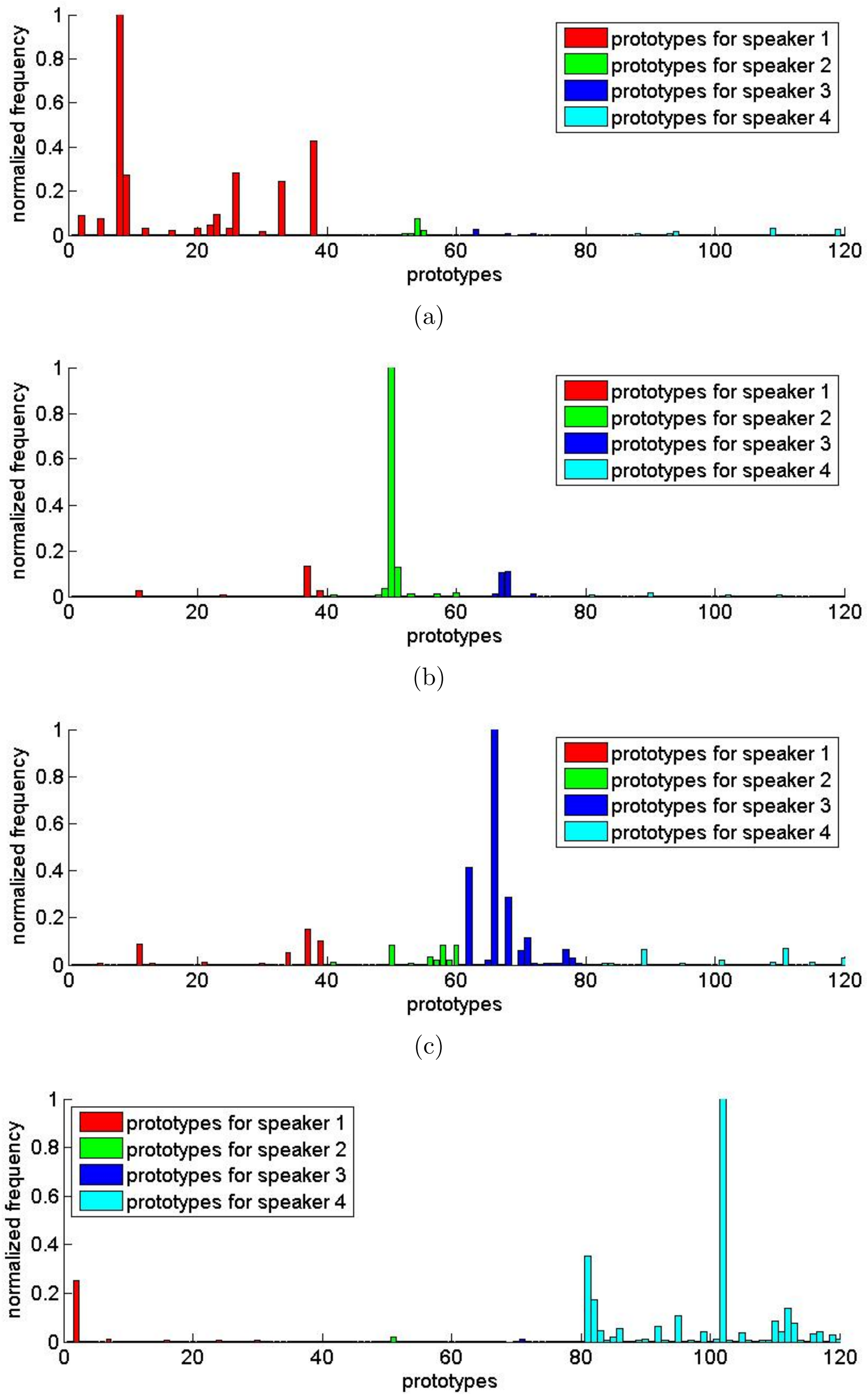
In the following, we use $Med2$ data to illustrate the process of our feature mapping. Fig. 6.8 displays the histograms of 4 segments that belong to different speakers before merging the prototypes. For this training data, when clustering the four speakers' segments (independently), we use $K = 40$, 20, 20, and 40 for speaker 1, 2, 3, and 4. respectively. The different number of clusters reflects the different number of training segments used for the four speakers. As it can be seen in Fig.6.8(a), the main response to a test sample from speaker 1 is in the first 40 prototypes that were learned from training data for this speaker. A similar behavior can be observed for test samples from speakers 2, 3, and 4 as shown in Fig.6.8(b)-(d).

Fig. 6.9 displays the histograms of the 4 input segments in Fig.6.8 but after merging similar prototypes and reducing the initial 120 prototypes to $K' = 100$ prototypes. As it can seen, after merging, speaker 1 still has high response to the first 27 prototypes, but also high response to prototypes 88 and 89. This is due to the similarity of these prototypes to those from class 1 that got deleted. Similarly, speaker 2 test segment has high response to the prototypes of speaker 2, and some combined prototypes.

Once each segment, i, is represented by a BoW feature, it can be classified using either a K-NN, SVM, or Naive Bayes (NB) classifier.

6.4.3 Fisher Vector Feature Mapping

Similar to the soft BoW feature mapping, Fisher Vector (FV) uses low-level features (e.g. MFCC, PLP, and LPCC) from each speech segment in the training set to learn the mapping. For this mapping, we use $K = 100$ Gaussian components.



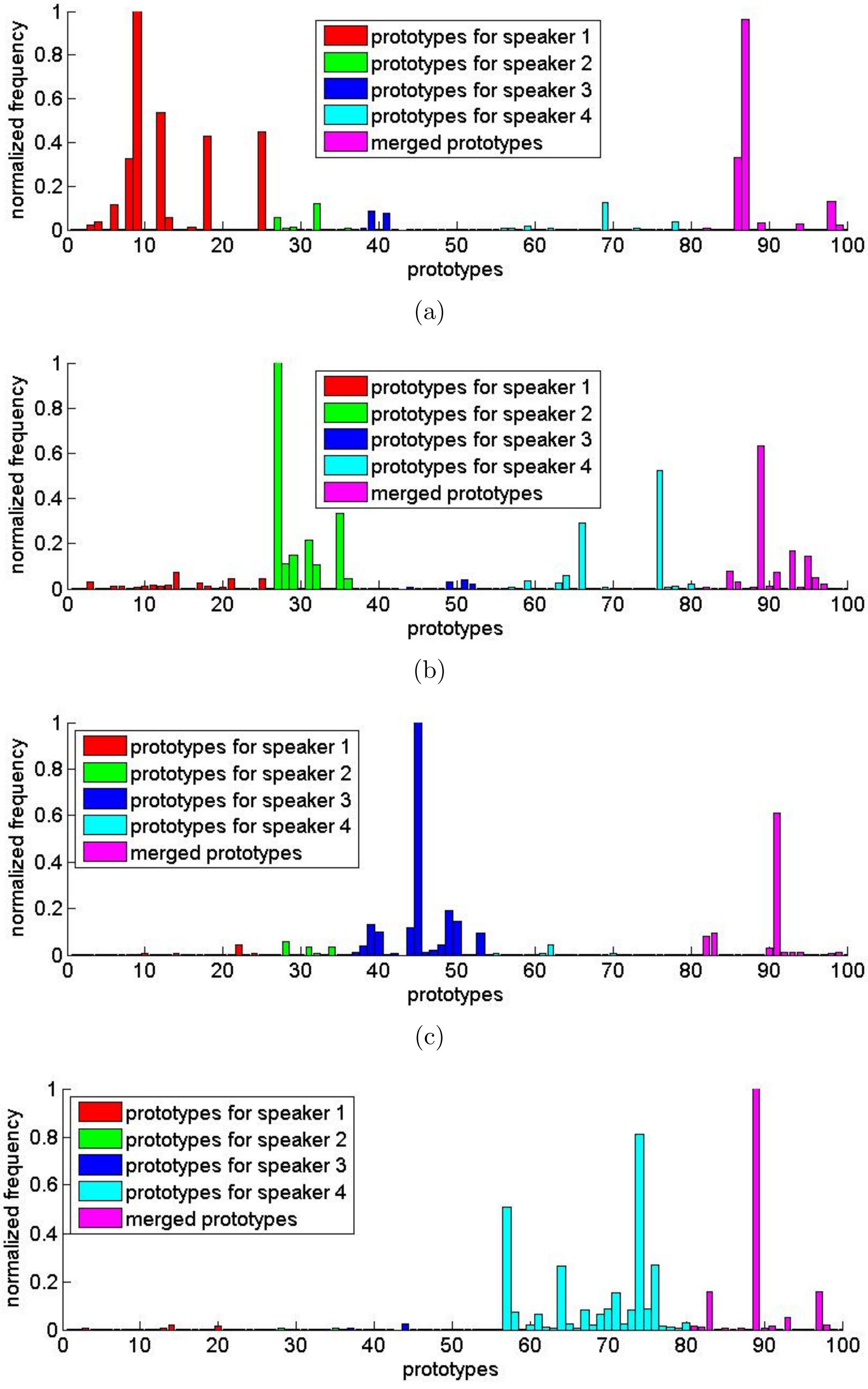
(a)

(b)

(c)

(d)

Figure 6.8: Response of all prototypes (before merging) to 4 segments from different speakers. (a) speaker 1, (b) speaker 2, (c) speaker 3, (d) speaker 4.



(a)

(b)

(c)

(d)

Figure 6.9: Response of all prototypes (after merging) to 4 segments from different speakers. (a) speaker 1, (b) speaker 2, (c) speaker 3, (d) speaker 4.

The parameters of the Gaussian components are learned from the training data using the EM algorithm. Then, for each speech segment $\boldsymbol{X}_t$, the FV feature representation, $g_\lambda^{\boldsymbol{X}_t}$ defined in (4.25) is calculated using equations (4.14)-(4.24). The dimensionality of the constructed FV for each segment is $100(Gaussian\ components) * (12(dimensions\ for\ mean)+12(dimensions\ for\ diagonal\ covariance)+1(component\ weight)) = 2500$.

Similar to BoW feature mapping, the FV feature representation also maps each speech segment to a 2500 dimensional feature vector regardless of the duration of the segment. Thus, a standard classifier, such as K-NN, or SVM, can be used to classify the mapped FV features of speech segments that have different sizes.

6.5 Speaker Segmentation

As described in chapter 3, we proposed two approaches to speaker segmentations. These are the extrema point-level fusion and the distance level fusion. In this chapter, these methods are evaluated using three independent speaker segmentation methods: T^2 [62], BIC [53], and $KL2$ [52] with 12 dimensional MFCC features. Similar to the $ChenBIC$ [53] algorithm, both fusion methods require the specification of 3 windows: initial window, growing window, an maximum window. Here, we fix them to 2sec, 1sec, and 12sec, respectively.

For the extrema point-level fusion, as illustrated in Section 3.2, changing points detected by the individual methods are merged together. This has a tendency to increase the detection of more true changing points, but also increase the number of false alarms. To reduce false alarms, we use a heuristic constraint that keeps only extrema points detected by at least 2 of the 3 speaker segmentation algorithms. Since different algorithms may identify the same extrema points at different locations, we assume that points detected within a 0.3 second from each other refer to the same change point.

The distance level fusion method assigns a weight coefficient, a_i, to each method i. We optimize these coefficients using cross-validation sets. Using the constraints that $a_i \in [0, 1]$ and $\sum_{i=1}^{3} a_i = 1$, we try all possible combinations of a_i with an increment of 0.1. We found that the values $a_1 = a_2 = 0.3$ and $a_3 = 0.4$ produce the best average performance across all validation datasets.

6.6 Speaker Identification

The proposed BoW and FV features were used to identify speakers using standard supervised learning with K-NN, SVM, and Naive Bayes (NB) classifiers.

We also use these features in a semi-supervised learning framework. In this case, we assume that only a limited amount of labeled data is available and we use both labeled and unlabeled data to build a classifier. We evaluate the proposed feature mappings within this framework as we vary the amount of labeled data. The unlabeled samples are labeled using label propagation approach [116] as described in Algorithm 5.1.

6.7 Emotion Recognition

Similar to speaker identification, the proposed BoW and FV feature representations are also applied to emotion recognition. The objective in this task is to identify the speech emotion regardless of the identity of the speaker.

Since our medical simulation data is small and has no ground truth for emotion classes, quantitative evaluation is not possible for this data. Thus, we use a existing emotion database (EMO-DB) to train and evaluate the emotion model using our proposed feature representations. The emotion recognition results for our medical simulation data can be evaluated qualitatively using our designed GUI.

6.8 Results and Analysis

6.8.1 Speaker Segmentation Algorithms Used for Comparison

To evaluate our proposed extrema point-level fusion (fusion-1) and distance level fusion (fusion-2) methods, five state-of-the-art speaker segmentation methods have been investigated and implemented for comparison purposes. These are Chen's BIC ($ChenBIC$) [53], sequential metric-based BIC ($SeqBIC$) [70], and three DAC-based methods ($DAC1$, $DAC2$, and $DAC3$) [71]. All algorithms were compared using the Mel-frequency cepstral coefficients (MFCC) [36], and the perceptual linear prediction(PLP) [41]. We have also experimented with the delta variations of these features [126].

All methods used for the comparison, including our proposed ones, are based on the Bayesian Information Criteria (BIC). The computation of the BIC requires the specification of a parameter, λ, (refer to equation (2.17)). In general, the results are sensitive to the selection of this parameter. Moreover, it is hard to fix it a priori. Thus, in our experiments, we vary $\lambda \in [0.5, 9]$ with a step of 0.1, and use the training data to identify the optimal value for this parameter.

6.8.1.1 Speaker Identification Algorithms Used for Comparison

In speaker identification experiments, we compare the performance of our proposed feature mapping methods: BoW (crisp, fuzzy, and possibilistic) and Fisher Vector to two state-of-the-art methods: GMM-UBM [99] and GMM mean supervector [127]. For all features, we use either the K-NN classifier (SV-KNN) or the SVM classifier (SV-SVM).

The GMM has been widely used [7, 37, 39, 78, 128] to represent the feature distribution of each speaker segment. In our experiemnts, the GMM model parameters, i.e. mean, covariance, and weight of each Gaussian component, are estimated using the EM algorithm [72]. Typically, more than 500 Gaussian components are needed to model the distribution of one speech segment. This large number is reasonable when the utterance is long (e.g. more than 3 minutes). However, when the segment is too short (e.g. less than 20 seconds), 500 components are too many to model the distribution of the features and may result in over-fitting. In fact, one Gaussian may be enough to represent one short utterance that is less than 5 seconds. Therefore, since a single number of Gaussian component needs to be fixed for all speech segments, choosing the number of GMMs can be a critical factor when constructing the GMM models.

GMM-UBM [99], is an adaptation of the GMM method. It uses all training data (all classes) to train a universal background model (UBM). Then, it iteratively adapts the universal model to each speaker utterance. All adapted models have the same number of components making this approach adequate to represent segments with different durations.

In the GMM-UBM approach, the similarity between a test utterance model (λ_{test}) and one training speaker model (λ_{C_i}) can be measured via the log-likelihood ratio [127], is defined as:

$$LLR(\mathbf{X}, \lambda_{test}, \lambda_{C_i}) = \frac{1}{T}\sum_{t=1}^{T}\{\log p(\boldsymbol{x}_t|\lambda_{test}) - \log p(\boldsymbol{x}_t|\lambda_{C_i})\} \tag{6.2}$$

In (6.2), $\mathbf{X} = \{\boldsymbol{x}_1, ..., \boldsymbol{x}_T\}$ are feature observations extracted from the T segments of the test sample, and $p(\boldsymbol{x}_t|\lambda)$ is the GMM density of observation $\boldsymbol{x}_t$.

The second approach used in our comparative analysis is the GMM mean supervector (SV) [127] feature representation method. SV is based on the GMM-UBM approach. Instead of using all Gaussian components to compute the log-likelihood ratio in (6.2). The SV uses only the Gaussian mean vectors. Specifically, the SV concatenates the mean vectors of all Gaussian components to create one high-dimensional feature vector. In our experiments, we use $K = 100$ Gaussian components with 12 dimensional MFCC, PLP, LPCC, and GFCC features. Thus, the dimension of SV feature vector is $12 * 100 = 1200$.

Similar to our proposed BoW and FV feature representation, the main advantage of the SV feature representation is that the low-level features are mapped to a fixed length feature vector regardless of the speech segment length. Thus, these features can be classified with standard classifiers such as K-NN or SVM.

6.8.1.2 Emotion Recognition

The proposed soft bag-of-words (BoW) and Fisher Vector (FV) feature representations are also used for the task of speaker emotion recognition. Emotion recognition and speaker identification are two different tasks that can complement each other. In speaker identification the objective is to distinguish the speaker's identity without considering its emotions. Similarly, in emotion recognition the objective is to identify the emotion of the speech segment regardless of the speaker's identity. As in the speaker identification experiments, we analyze and compare our BoW and FV feature representations with the GMM mean supervector (SV) [129] for emotion recognition.

6.8.2 Evaluation Measures

6.8.2.1 Speaker Segmentation

We use several measures to analyze and compare the performance of speaker segmentation algorithms. Two such measures are the false alarm rate (FAR) and the misdetection rate (MDR) [44, 47, 70]. These measures are defined as

$$FAR = \frac{FA}{GT + FA}, \tag{6.3}$$

and

$$MDR = \frac{MD}{GT}. \tag{6.4}$$

In (6.3) and (6.4), FA denotes the number of false alarms, MD denotes the number of misdetections, and GT stands for the actual number of speaker change points, i.e. the ground truth. A false alarm occurs when a false speaker change point is detected. A misdetection occurs when an actual speaker change point is not detected by the algorithm.

Another category of performance measures are based on precision (PRC), recall (RCL), and $F1$ measure [57, 130]. These measures are defined as

$$PRC = \frac{CFC}{DET} = \frac{CFC}{CFC + FA}, \tag{6.5}$$

$$RCL = \frac{CFC}{GT} = \frac{CFC}{CFC + MD}, \tag{6.6}$$

and

$$F1 = 2 * \frac{PRC * RCL}{PRC + RCL}. \tag{6.7}$$

In the above, CFC denotes the number of correctly detected change points and DET denotes the total number of the detected speaker change points, i.e. $DET = CFC + FA$.

We should note that the pair of measures (FAR, MDR) and (PRC, RCL) hold the following relationships:

$$MDR = 1 - RCL, \tag{6.8}$$

and

$$FAR = \frac{RCL * FA}{DET * PRC + RCL * FA}. \tag{6.9}$$

6.8.2.2 Speaker Identification

For all experiments for speaker identification, we use k-fold cross validation with k = 5. That is, for each video, we keep 80% of data for training and use the remaining 20% for testing. We repeat this process 5 times by testing different subsets and report the average classification rate of the 5 runs.

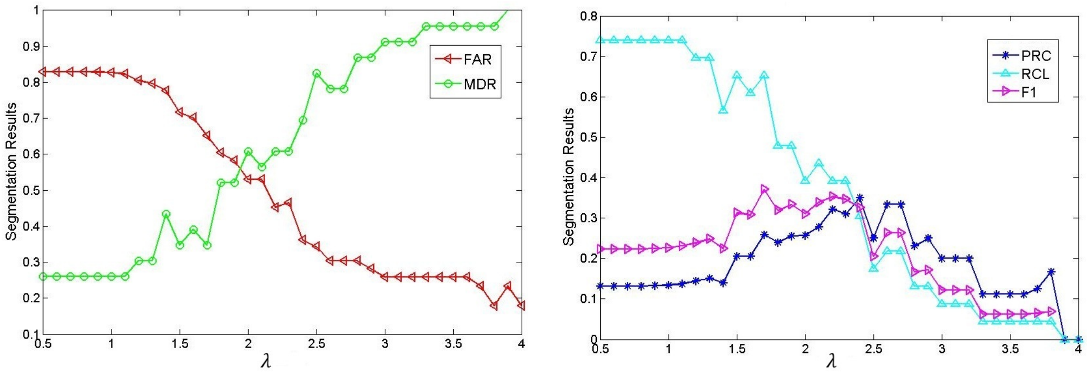
We should note that each video segment is processed independently since it involves different speakers. The reported results are the average over the 15 datasets.

6.8.3 Results and Discussion

6.8.3.1 Speaker Segmentation

We use the $Med2$ data as an illustrative example for speaker segmentation. Then, we report results on all simulation videos. First, the entire audio sequence is down-sampled into $fs = 22050Hz$ wave signal and decomposed into small (25ms) analysis window frames with 10ms overlap. The dimension of the MFCC features is set to 12. Thus, for the 7m13s audio data in $Med2$ video, the MFCC features correspond to a 43298×12 dimensional matrix.

Fig. 6.10 displays the results obtained by the $ChenBIC$ segmentation algorithm on $Med2$ data when λ is varied from 0.5 to 4 with a step of 0.1. Fig. 6.10(a) shows the FAR and MDR measures as a function of λ. As it can be seen, a low λ results in a high false alarm rate and a low misdetection rate. As we increase λ, these two measures move in the opposite direction. That is, the FAR decreases while the MDR increases. Fig. 6.10(b) illustrates the behavior of the other three measurements (PRC, RCL, and $F1$). For these measures, a low λ results in high recall but low precision. As we increase λ, the recall drops and the precision increases. The precision ceases to increase as λ is increased beyond 2.5. Fig. 6.11 shows a similar behavior for the other four segmentation algorithms ($SeqBIC$, $DAC1$, $DAC2$, and $DAC3$).

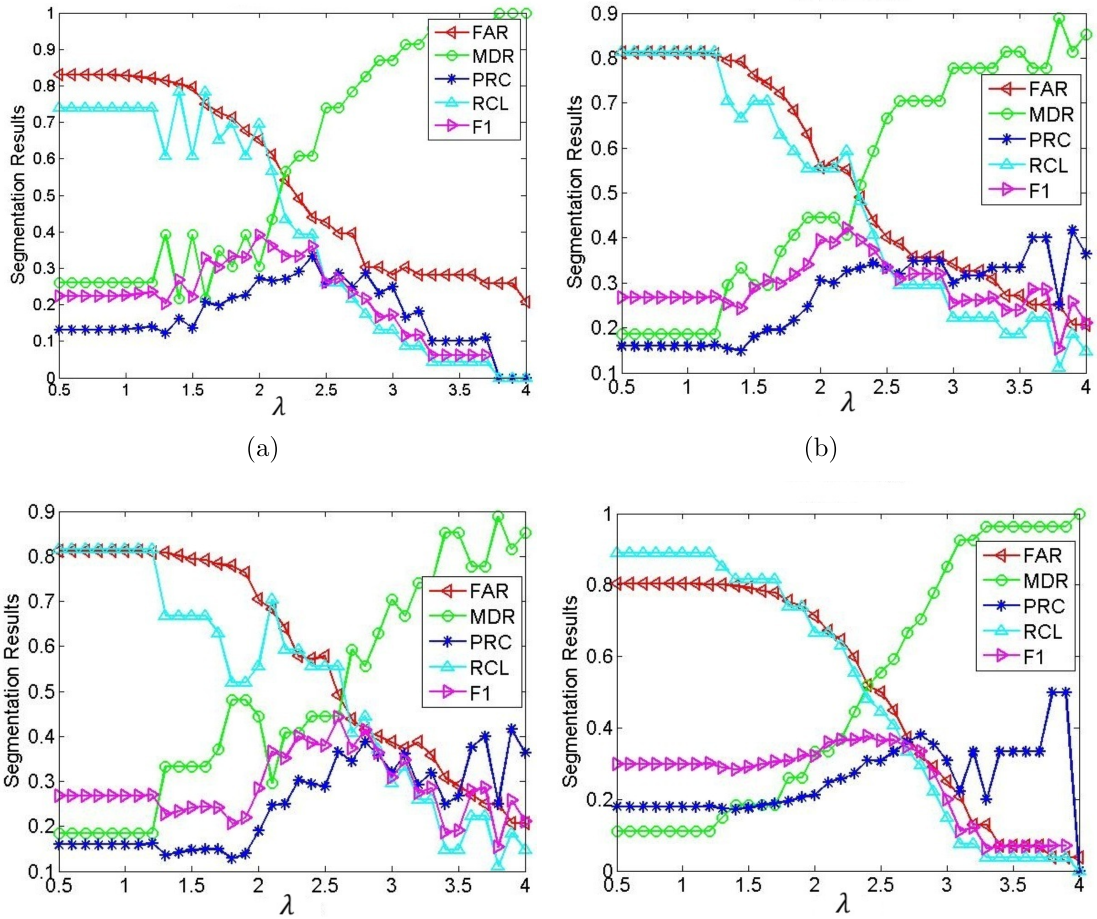


(a) (b)

Figure 6.10: Evaluation measures of $ChenBIC$ speaker segmentation method on $Med2$ audio data with 12dim MFCC features.

The optimal value of λ depends on the cost of false alarms and the cost of misdetection. In speaker segmentation, it is better to have an over-segmentation than an under-segmentation. This is because over-segmentation (i.e. large FAR) is tolerable and can potentially be fixed in subsequent steps. In particular, two adjacent segments around a false changing point can be identified in a post-processing step as belonging to the same speaker class. Thus, false alarms may not have a significant effect on the final recognition results. On the other hand, the cost of misdetection is much higher. A misdetection can result in one segment containing speech of multiple speakers. The features extracted from this segment would combine and average the characteristics of multiple speakers. Consequently, this type of error cannot be fixed in post-processing and would result in misclassification in the speaker identification or emotion recognition step. Based on this analysis, we select the value of λ that minimizes the misdetection. Using the results displayed in Fig. 6.10 and 6.11, we let $\lambda = 1$ for the 12 dimension MFCC-based segmentation algorithms.

Fig. 6.12 shows a more detailed analysis of the results of $ChenBIC$'s segmen-



(a) (b)

(c) (d)

Figure 6.11: Evaluation measures of four speaker segmentation methods on $Med2$ audio data with 12dim MFCC features. (a) $SeqBIC$ method, (b) $DAC1$ method, (c) $DAC2$ method, (d) $DAC3$ method.

tation algorithm on $Med2$ audio data. This figure compares the detected speaker change points to the location of the 27 actual change points. The algorithm detected a total of 136 change points, only 20 of these are true changes. The remaining 116 points are false alarms. These false alarms would result in an over-segmentation where a large number of segments will be fed to the classification step.

Fig. 6.13 shows a more detailed analysis of the results of the $DAC3$-based detection algorithm which has the best overall performance. This method detected

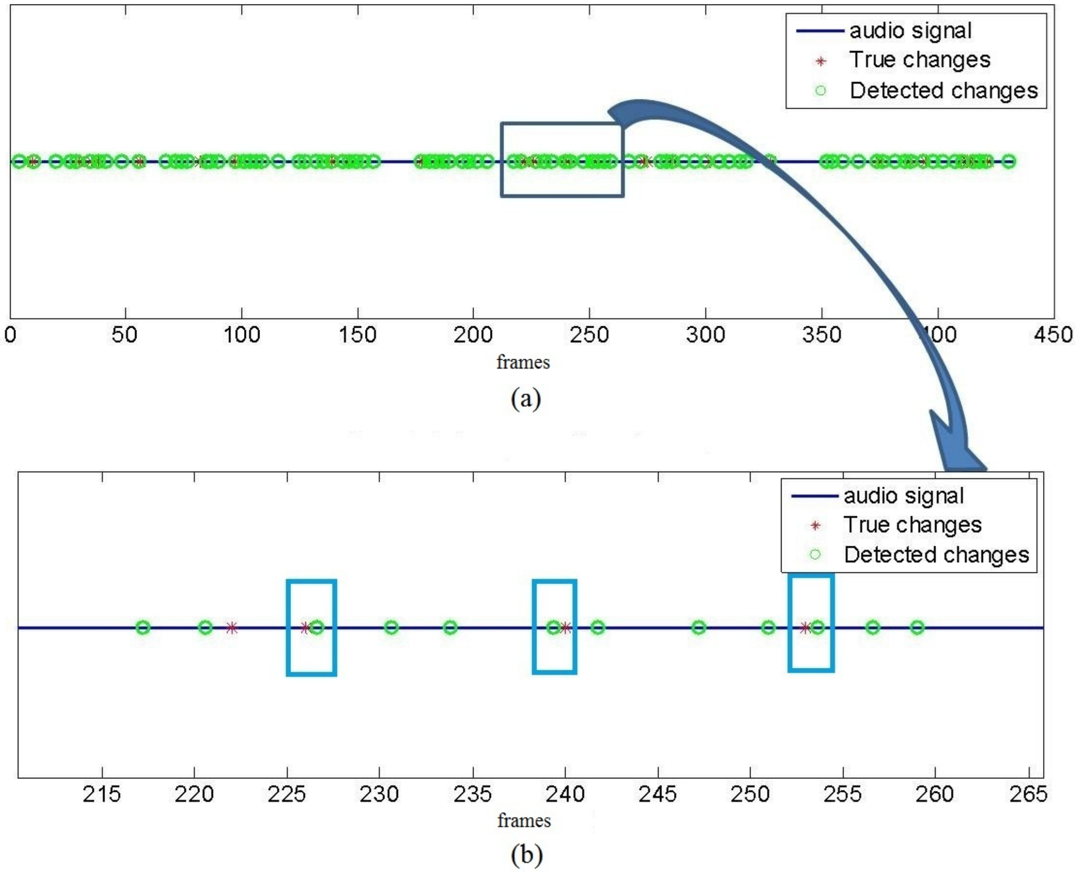
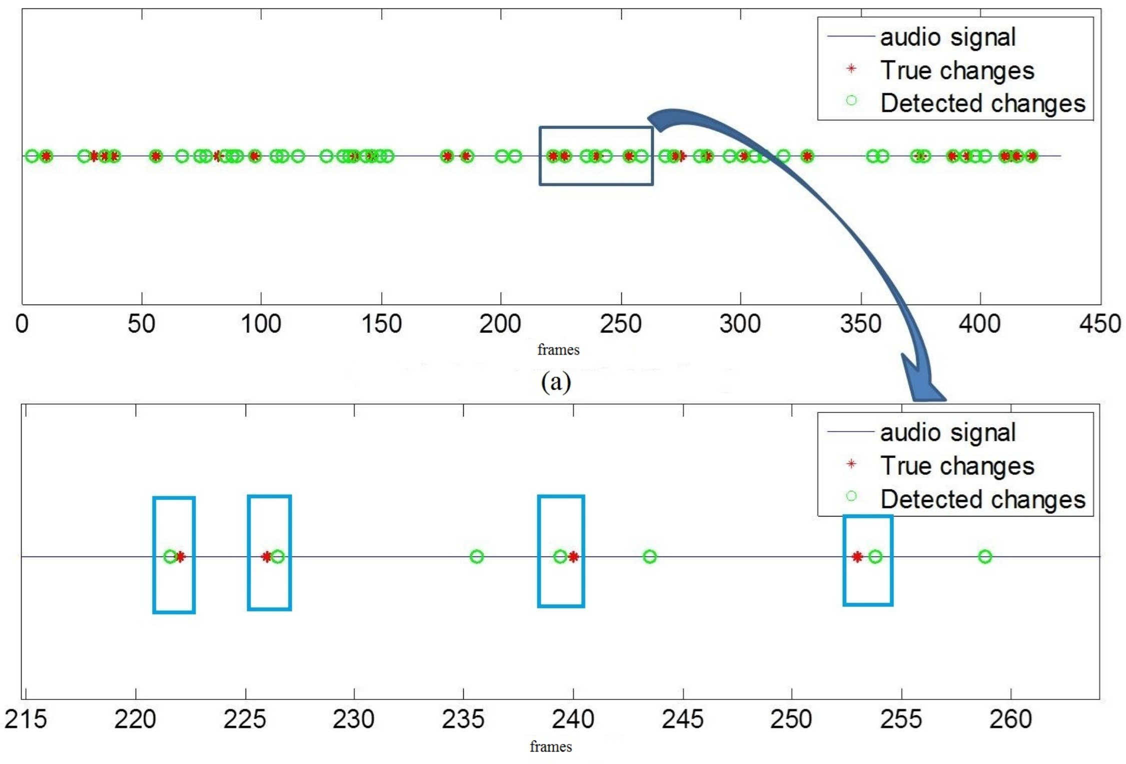


Figure 6.12: Detected changing points by $ChenBIC$ and real changing points in $Med2$ audio data. (a) results on the entire audio signal, (b) details of the results from 215s to 265s.

128 changing points with 24 real ones and 104 false alarms.

All of the BIC-based segmentation algorithms that we analyzed are flexible and can integrate various features with various dimensions. So far, we have only compared them using 12-dimensional MFCC features. In the following, we analyze the performance of the $DAC3$ algorithm (the one that has the best performance) with the MFCC, PLP, and their Δ and $\Delta\Delta$ features with various dimensions. Fig. 6.14 shows the DAC3-based segmentation evaluation using the MFCC features with differ-

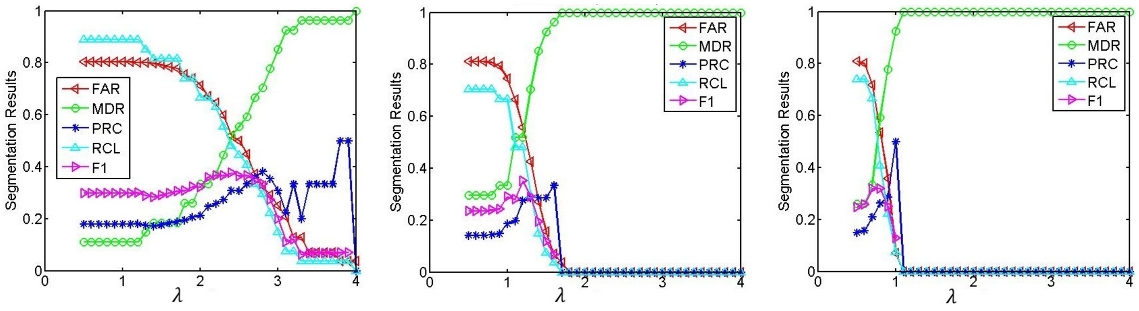


(a)

(b)

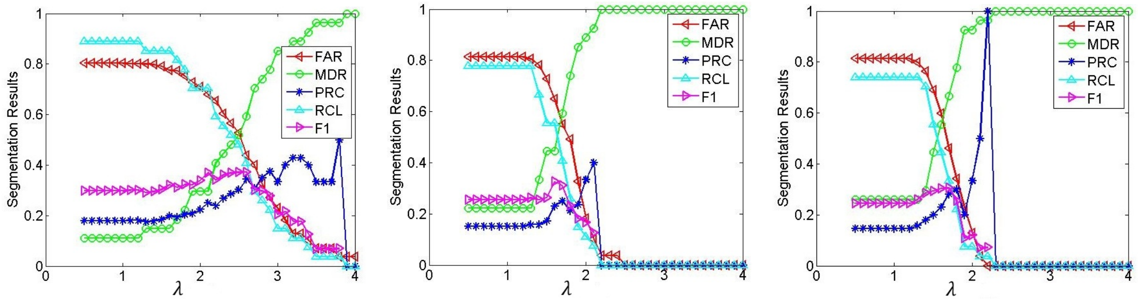
Figure 6.13: Detected changing points by $DAC3$ and real changing points in $Med2$ audio data. (a) results on the whole audio signal, (b) details of the results of the audio segment from 215s to 265s.

ent dimensions. Fig. 6.15 shows the DAC3-based segmentation results using the PLP features with different dimensions. For both features, the parameter λ is varied from 0.5 to 4 with a step of 0.1. As it can be seen, the performance of both segmentation algorithms decreases as the dimensionality of the features increases. For instance, when $\lambda = 0.7$, the RCL is dropped from 89% to 50% and the MDR is increased from 12% to 27% when the dimensionality of the MFCC features is increased from 12 to 36. Similar results were observed with the other segmentation algorithms. Thus, we can conclude that increasing the dimension of the MFCC or PLP would decrease the performance of the speaker segmentation algorithm.



(a) (b) (c)

Figure 6.14: Evaluations of DAC3 speaker segmentation method on $Med2$ audio data using MFCC features with (a) 12 dimension, (b) 24 dimension, and (c) 36 dimension.

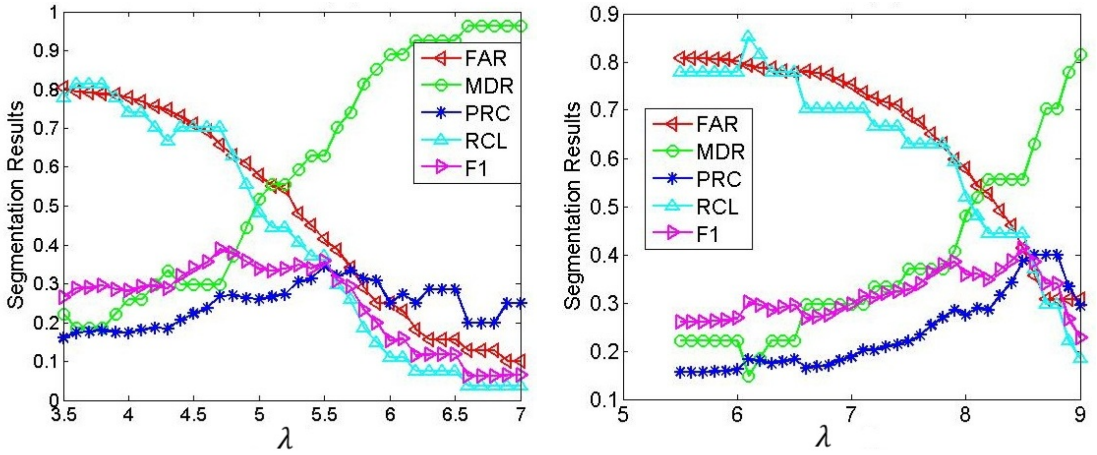


(a) (b) (c)

Figure 6.15: Evaluations of DAC3 speaker segmentation method on $Med2$ audio data using PLP features with (a) 12 dimension, (b) 20 dimension, and (c) 23 dimension.

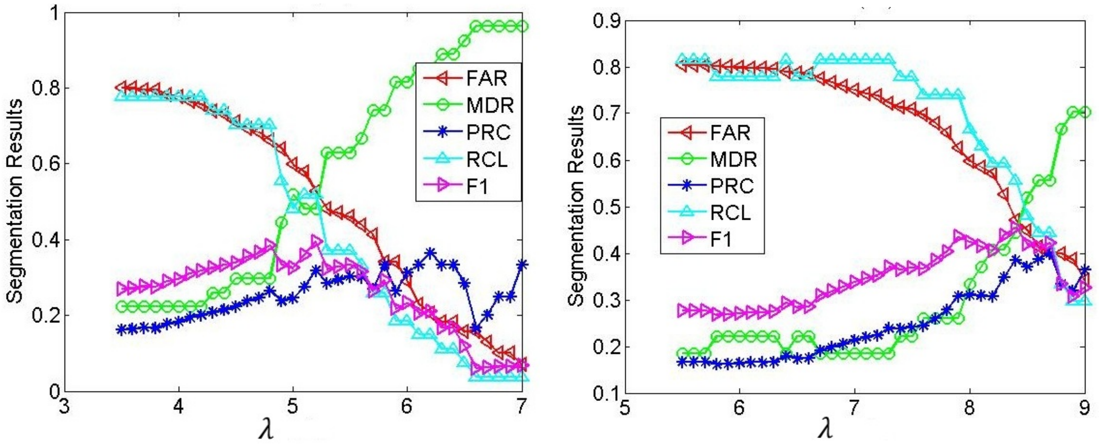
Fig. 6.16 compares the DAC3 segmentation results when the Δ and $\Delta\Delta$ features are added to the 12-dimensional MFCC features. Similarly, Fig. 6.17 compare the results using the PLP and its Δ and $\Delta\Delta$ features. For this experiment, we vary λ from 3 to 7 with a step of 0.1. This large range of values is needed due to the larger number of features. As it can be seen, adding the derivative of MFCC or PLP features did not improve the performance of either segmentation algorithms. In fact, it may decrease the RCL. This is in addition to the extra computation needed for these extra features. From Fig. 6.14(a), 6.15(a), we can also conclude that the MFCC and PLP features generate comparable results.

From the above experiments, we can see that all methods have high FAR val-



(a) (b)

Figure 6.16: Evaluations of DAC3 based speaker segmentation methods on $Med2$ audio data with (a) MFCC(12dim)+Δ, and (b) MFCC(12dim)+Δ+ΔΔ features.



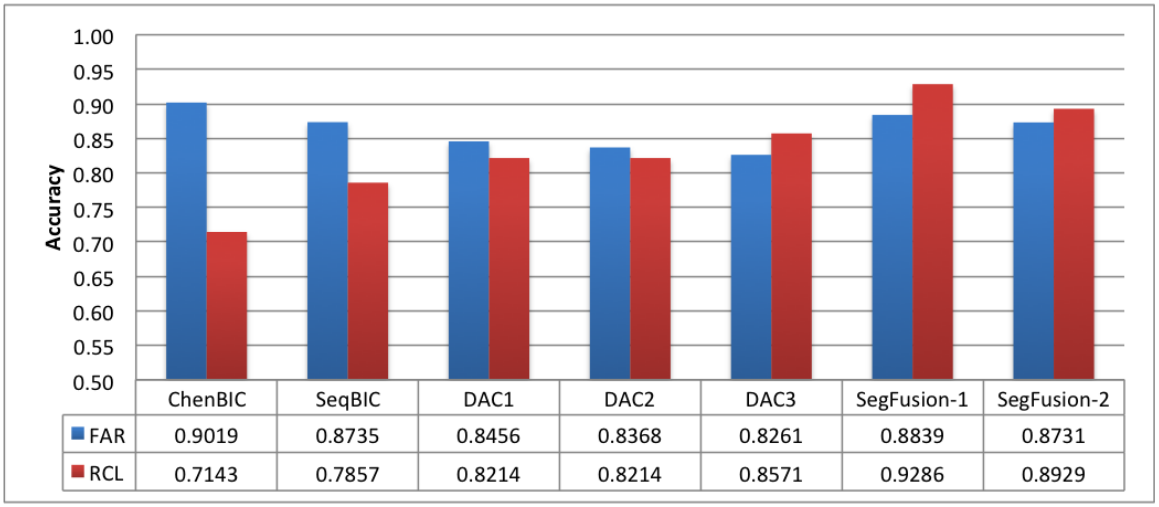
(a) (b)

Figure 6.17: Evaluations of DAC3 based speaker segmentation methods on $Med2$ audio data with (a) PLP(12dim)+Δ, and (b) PLP(12dim)+Δ+ΔΔ features.

ues resulting in low PRC and $F1$ values. Thus, the overall segmentation results are not accurate. This poor performance is mainly due to the low quality of the audio recording and the noisy background. We can also conclude that the $DAC3$ algorithm provided the best segmentation results among the five considered methods. The challenge in speaker segmentation is to detect all speaker changes while keeping the false alarm rate as low as possible. As we have argued, the cost of a misdectection is much higher than the cost of a false alarm. Thus, we should first aim to detect all possible speaker changing points, and then try to reduce the false alarms as much as possible without degrading the detection rate.

For the two fusion methods, extrema point-level fusion and distance level fusion, proposed in chapter 3, we first implement the segmentation approaches based on BIC [53], KL [52] and T^2 [62] respectively. Then, we use the proposed fusion methods to combine the results of the three algorithms.

Fig. 6.18 compares the FAR and RCL performance measures of the seven segmentation algorithms on $Med2$ dataset where $\lambda = 1$. As it can be seen, extrema point-level fusion (SegFusion-1) has the best results with the highest RCL (92.86%), while distance level fusion (SegFusion-2) obtains 89.29% RCL and a little lower FAR, DAC-3 has lower RCL (85.7%) but also lower FAR than the two fusion methods.



	ChenBIC	SeqBIC	DAC1	DAC2	DAC3	SegFusion-1	SegFusion-2
🟦 FAR	0.9019	0.8735	0.8456	0.8368	0.8261	0.8839	0.8731
🟥 RCL	0.7143	0.7857	0.8214	0.8214	0.8571	0.9286	0.8929

Figure 6.18: Comparison of the FAR (first bar) and RCL (second bar) of seven speaker segmentation methods on $Med2$ audio data with 12dim MFCC features when $\lambda = 1$.

Table 6.2 reports the results of the seven speaker segmentation algorithms with MFCC features on all 15 data sets. For each algorithm and each data set, we display the number of detected changing points and the number of true changes (in $(\cdot)$). For instance, for *Med1*, 204 speaker changes were detected by extrema point-level fusion and only 24 of them are true changes. As it can be seen, the over-segmentation problem is an issue for all algorithms. The proposed extrema point-level fusion results in a larger number of segments than the other methods. This is expected since it considers three metrics, BIC, KL, and T^2 simultaneously. Distance level fusion has a slightly fewer number of segments than the extrema point-level fusion. This is due to the averaging/weighting of each metric.

TABLE 6.2

Speaker segmentation results by 7 different algorithms. For each algorithm, we report the number of detected changing points and the number of true changes in ().

Videos	Lengths	# of Speakers	# True changes	ChenBIC	SeqBIC	DAC1	DAC2	DAC3	Extrema point-level	Distance level fusion
Med1	6m35s	4	24	198 (17)	177 (18)	152 (20)	140 (22)	132 (22)	204 (24)	201 (24)
Med2	7m13s	4	28	204 (20)	174 (22)	149 (23)	141 (23)	136 (24)	224 (26)	197 (25)
Med3	10m20s	4	31	216 (23)	188 (24)	164 (24)	158 (25)	153 (26)	235 (30)	221 (29)
Med4	18m02s	4	43	280 (30)	279 (32)	277 (33)	261 (35)	250 (36)	326 (39)	289 (37)
Med5	9m40s	4	29	237 (22)	238 (23)	204 (24)	197 (24)	189 (25)	245 (28)	233 (26)
Med6	7m22s	4	34	160 (26)	179 (26)	178 (28)	164 (28)	147 (29)	168 (32)	154 (28)
Med7	10m16s	4	29	245 (21)	248 (23)	243 (23)	241 (24)	222 (25)	253 (28)	242 (23)
Med8	4m33s	4	22	82 (16)	87 (17)	79 (17)	85 (18)	79 (19)	94 (21)	77 (20)
Med9	6m54s	4	31	102 (22)	108 (22)	99 (24)	105 (25)	97 (25)	125 (29)	108 (26)
Med10	5m32s	4	18	110 (13)	113 (14)	94 (15)	93 (14)	86 (16)	129 (17)	105 (16)
Med11	6m43s	4	26	107 (18)	119 (19)	104 (20)	102 (19)	98 (22)	116 (24)	100 (22)
Med12	7m45s	4	22	236 (18)	247 (17)	203 (18)	207 (18)	198 (19)	246 (21)	219 (18)
Med13	12m1s	4	32	259 (24)	289 (24)	233 (26)	238 (25)	227 (27)	271 (29)	249 (27)
Med14	5m34s	4	17	112 (13)	139 (13)	111 (13)	115 (14)	101 (14)	125 (15)	118 (13)
Med15	7m55s	4	37	163 (26)	171 (26)	165 (28)	162 (28)	155 (30)	190 (33)	167 (29)

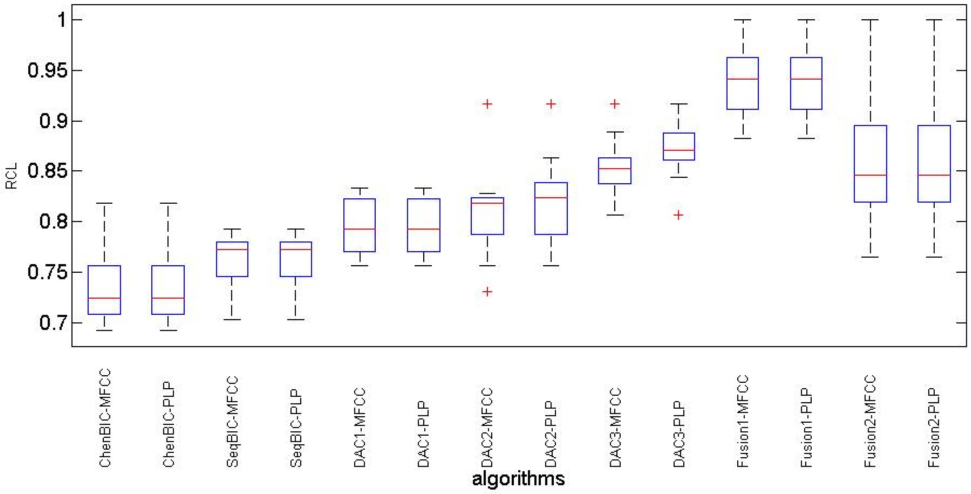


Figure 6.19: Statistics of the RCL measure values over the 15 data sets of the seven methods with MFCC and PLP features. For each algorithm, the box represents the statistics of the values, the red line is the median, the edges of the box are the 25th and 75th percentiles.

Figure 6.19 summarizes the statistics of the RCL measure over the 15 data sets of the seven segmentation methods. As it can be seen, extrema point-level fusion (Fusion1) obtains a higher RCL rate than other methods (about 8% percent improvement). The performance of the distance level fusion (Fusion2), on the other hand, is not consistent over all 15 data sets.

Figure 6.20 displays the statistics of the FAR measure over the 15 data sets of the seven segmentation methods. As it can be seen, all methods have a large number of false alarms. This is due mainly to the noisy nature of the data (as justified earlier). Also, there is a significant variation of the FAR among the different data sets. The extrema point-level fusion has a slightly higher FAR rate than most methods.

By identifying more segments, it is more likely that more of the true changing

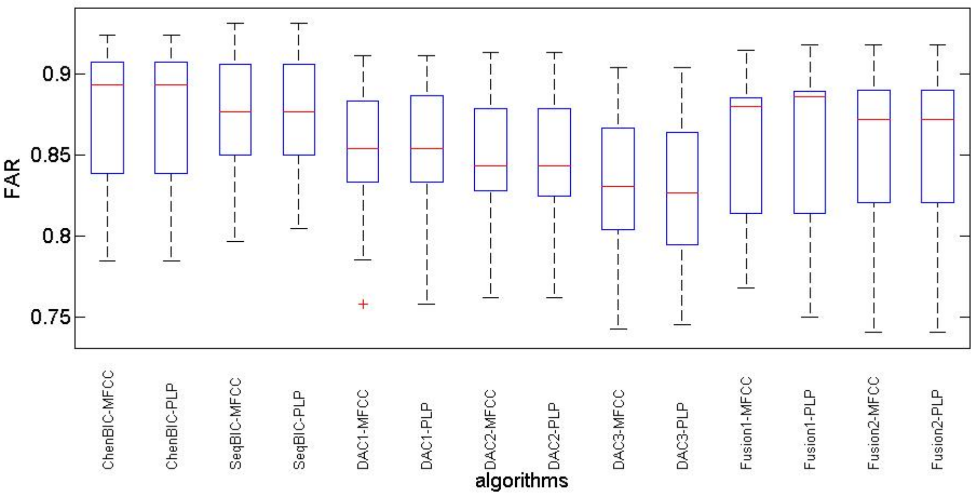


Figure 6.20: Statistics of the FAR measure values over the 15 data sets of the seven methods with MFCC and PLP features.

points can be detected. As we have argued earlier, the cost of a misdetection is much higher than the cost of a false alarm. In other words, subsequent processing steps can remedy the over-segmentation caused by false alarms but not the misdetected changing points. Thus, the proposed extrema point-level fusion is a better choice for the speaker segmentation task.

In the above analysis, seven speaker segmentation methods were implemented and compared. Several variations of the MFCC and PLP features were extracted from audio after classification-based silence removal. Most segmentation algorithms produce an acceptable misdetection rate at the expense of a high false alarm rate. The performance of all algorithms is highly dependent on the quality of the recording. For the single metric, the $DAC3$ segmentation algorithm has the best overall performance. Our proposed two fusion approaches, extrema point-level fusion and distance level fusion, have also achieved promising results. In the next sections, we use these segmented results of the extrema point-level fusion to perform subsequent speaker identification and emotion recognition steps.

6.8.3.2 Speaker Identification

First, the proposed bag of words features (C-BoW, F-BoW, and P-BoW), as described in Section 4.2, are constructed from four different low-level features, i.e. MFCC, PLP, LPCC, and GFCC. Then, we evaluate and compare their performance based on the K-NN classifier.

For the K-NN classifier, first we experiment with several measures to compute the dissimilarity between two histogram features (i.e. vectors mapped to histograms using bag of words representation). In particular, we use chi-square statistics (CS), histogram intersection (HI), Jensen-Shannon divergence (JS), Kolmogorov-Smirnov distance (KS), Kullback-Leibler divergence (KL), match distance (MD), diffusion distance (DD), and cosine distance (CD). The speaker recognition accuracies, averaged over the 15 datasets, using the MFCC features with a K-NN classifier (K=7), are displayed in Table 6.3. As it can be seen, the cosine distance (CD) has the best performance for the crisp, fuzzy, and possibilistic bag of words representations. Similar results are obtained for the PLP, LPCC, and GFCC features, as well as for the proposed FV feature representation method. Thus, for the remaining experiments, the cosine distance will be used within the K-NN classifier to compare our features to other classifiers and features.

In a second experiment, we compare the speaker identification accuracy of the proposed soft BoW feature mappings using MFCC features with the K-NN, NB, and

TABLE 6.3

Classification rate of the K-NN classifier using the proposed soft bag of words representation of MFCC features and various distance measures

Dist. Type	C-BoW	F-BoW	P-BoW
Eu	0.756	0.775	0.77
CS	0.742	0.766	0.758
HI	0.752	0.765	0.752
JS	0.545	0.571	0.552
KS	0.792	0.809	0.799
KL	0.555	0.573	0.564
MD	0.793	0.808	0.803
DD	0.715	0.734	0.739
CD	**0.794**	**0.816**	**0.806**

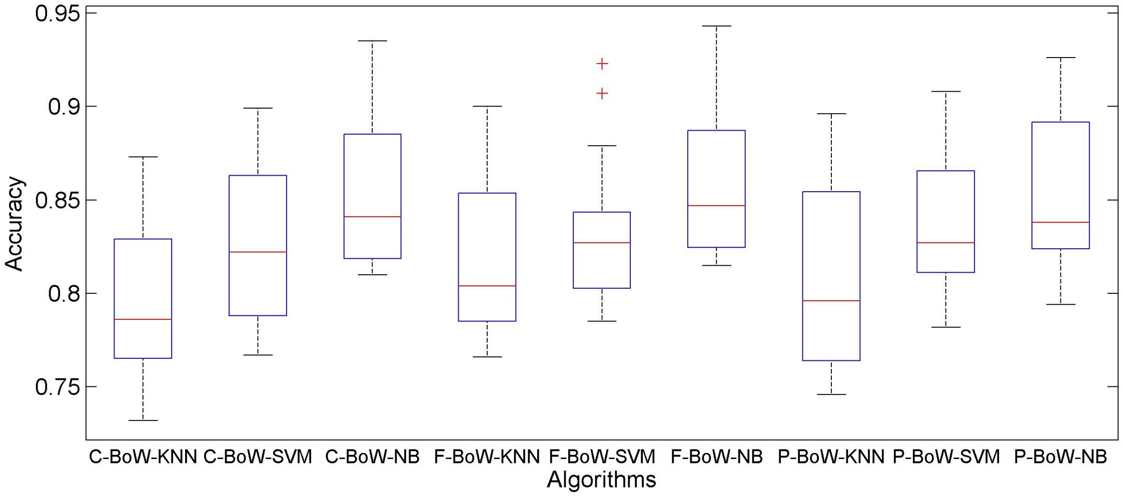


Figure 6.21: Performance of the crisp, fuzzy, and possibilistic BoW using MFCC features with the KNN, SVM, and NB classifiers

SVM classifiers. The results are reported in Figure 6.21. First, we notice that the NB classifier outperforms the K-NN and SVM classifiers for the crisp, fuzzy, and possibilistic cases. Second, on average, the soft (fuzzy and possibilistic) feature mappings outperform the crisp mapping. Similar results were obtained for the PLP, LPCC, and GFCC features.

In a third experiment, we evaluate the performance of the proposed FV rep-

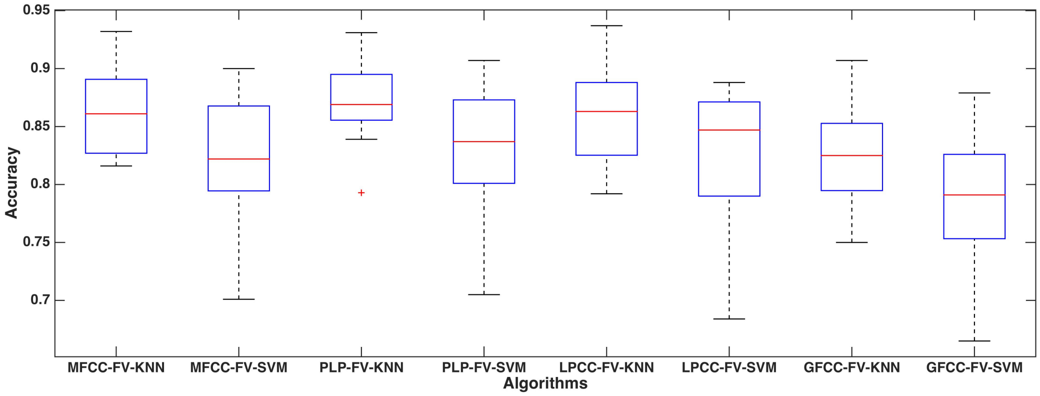


Figure 6.22: Performance of the Fisher Vector representation using MFCC, PLP, and LPCC features with the KNN and SVM classifiers

resentation for each extracted feature using two types of classifiers: K-NN, and SVM [131]. We report the results of the K-NN with the cosine distance, and SVM with linear kernel. For each classifier, we compare the performance of the MFCC, PLP, and LPCC based FV feature representation methods. The results are reported in Figure 6.22. As it can be seen, the K-NN classifier outperforms the SVM linear kernel classifier for all three features

In a fourth experiment, using the best settings for our methods (BoW with NB classifier and FV with KNN) and compare them to existing speaker identification algorithms: GMM-UBM [99], GMM mean supervector [127] with K-NN classifier (SV-KNN) and SVM classifier (SV-SVM), as described in section 6.8.1.1. The results for different low-level features are reported in Figures 6.23 - 6.26. As it can be seen, for all 4 features, both soft feature mapping coupled with the NB classifier and Fisher Vector with KNN classifier outperform the state of the art methods. The FV features have a slight improvement over the fuzzy and possibilistic BoW. The p-value between FV-KNN and SV-KNN is 0.0002, which is much smaller than 0.05, indicating the significant improvement for our proposed method.

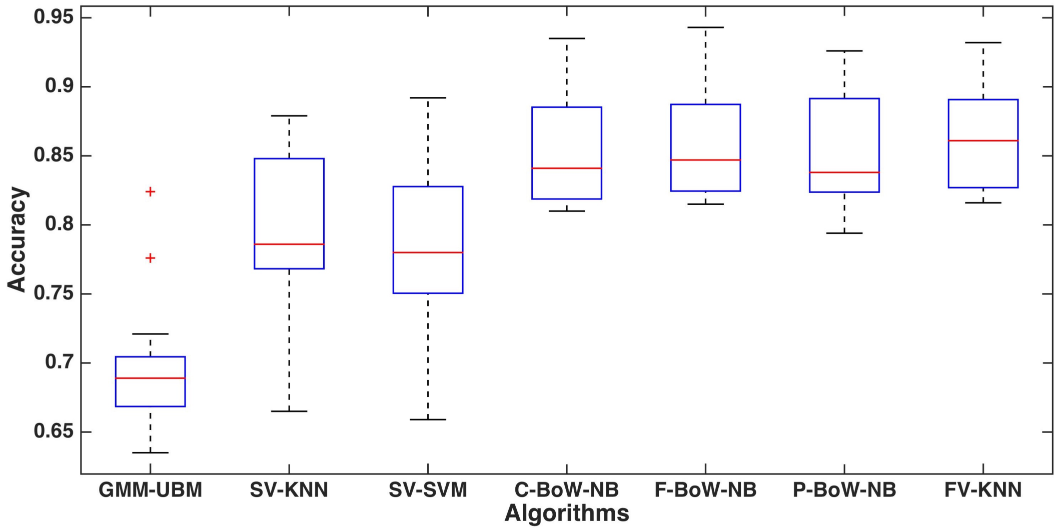


Figure 6.23: Comparison of the classification accuracy of soft BoW feature mappings (C-BoW, F-BoW, and P-BoW) using the NB classifier and FV feature mapping with KNN classifier with GMM-UBM, GMM mean supervector with K-NN (SV-KNN) and SVM (SV-SVM) using MFCC features. The results are averaged over 15 datasets.

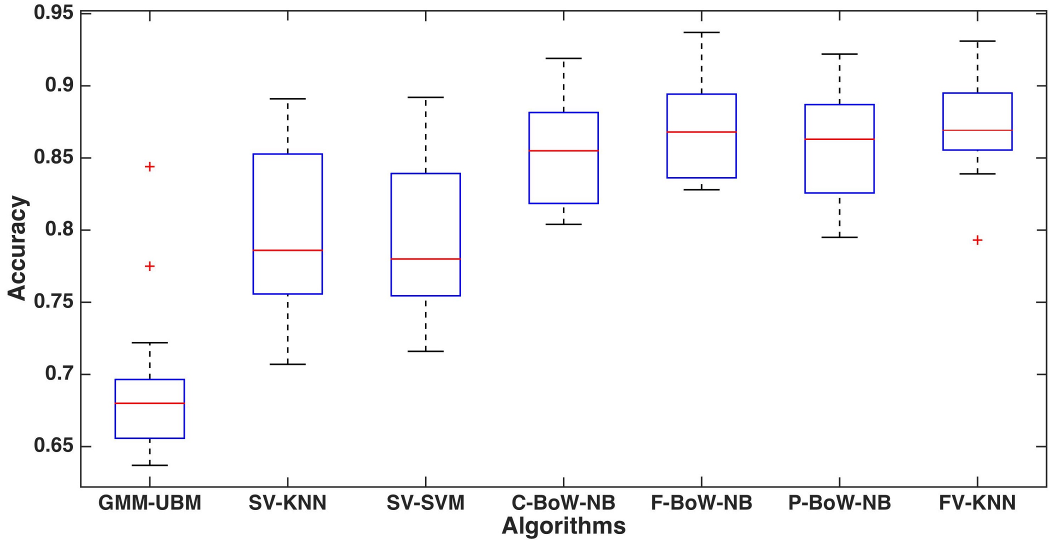


Figure 6.24: Comparison of the classification accuracy of soft BoW feature mappings (C-BoW, F-BoW, and P-BoW) using the NB classifier and FV feature mapping with KNN classifier with GMM-UBM, GMM mean supervector with K-NN (SV-KNN) and SVM (SV-SVM) using PLP features. The results are averaged over 15 datasets.

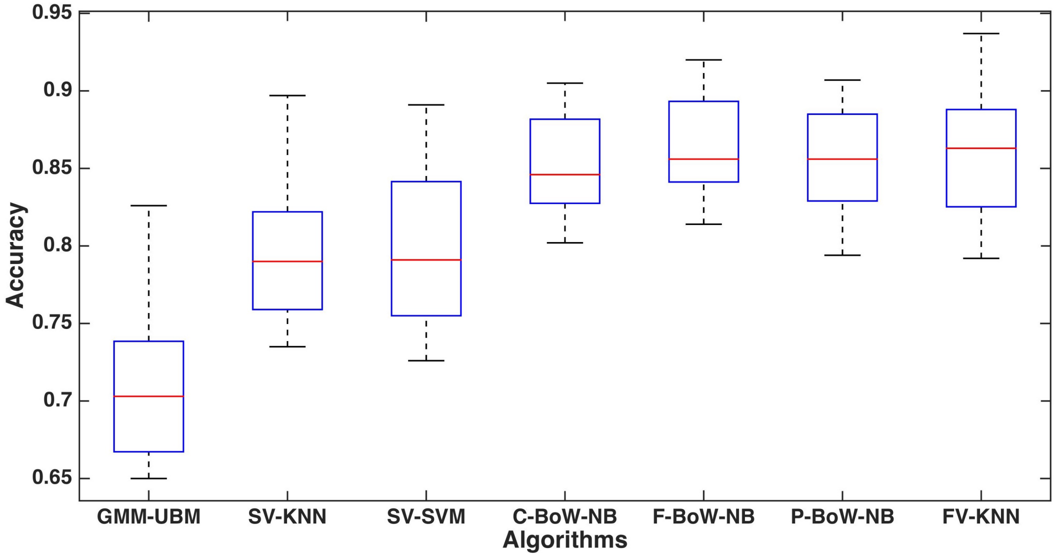


Figure 6.25: Comparison of the classification accuracy of soft BoW feature mappings (C-BoW, F-BoW, and P-BoW) using the NB classifier and FV feature mapping with KNN classifier with GMM-UBM, GMM mean supervector with K-NN (SV-KNN) and SVM (SV-SVM) using LPCC features. The results are averaged over 15 datasets.

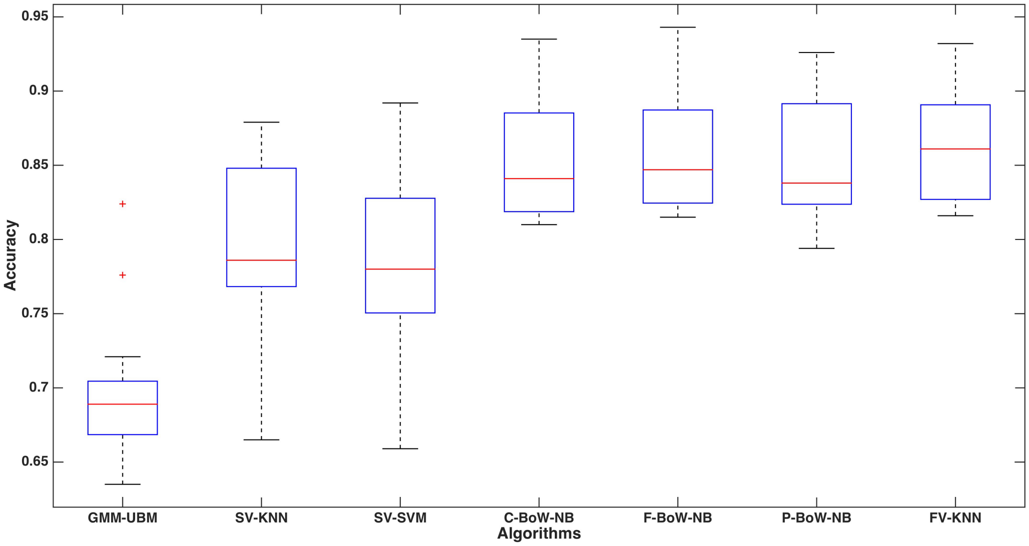


Figure 6.26: Comparison of the classification accuracy of soft BoW feature mappings (C-BoW, F-BoW, and P-BoW) using the NB classifier and FV feature mapping with KNN classifier with GMM-UBM, GMM mean supervector with K-NN (SV-KNN) and SVM (SV-SVM) using GFCC features. The results are averaged over 15 datasets.

From Figures 6.23 - 6.26, we also note that the classification results of all algorithms have large standard deviations. This means that the classification rates are high for some data sets and low for others. This is because the performance of all algorithms is highly dependent on the quality of the audio and the pre-segmentation results. For some data sets, the accuracy rates can be over 90%. This is because these simulations include both male and female speakers (it is relatively easier to discriminate between speakers of different gender). Additional factors that can yield accurate speaker identification include: (1) higher recording quality, (2) more clear pronunciation, (3) better segmentation results.

The analysis of the mis-classified speaker segments shows that all methods fail when the segment contains multi-speakers. We have also observed that some segments are correctly classified by our BoW-based methods while misclassified by the GMM-UBM/SV-based method. These are typically very short segments where the data is not sufficient to estimate the GMM components efficiently.

Our results have also indicated that PLP features provide a slightly better discrimination than the MFCC, LPCC, or GFCC features.

6.8.3.3 Semi-supervised Speaker Identification

In the previous section, we reported the results of various speaker identification methods that use a standard supervised learning approach. In these methods, 80% of the data were labeled and used to train a classifier. The remaining 20% were assumed unlabeled and used to test the classifier. In this section, we report the results of using the semi-supervised learning algorithm described in Section 5.3. For this experiment,

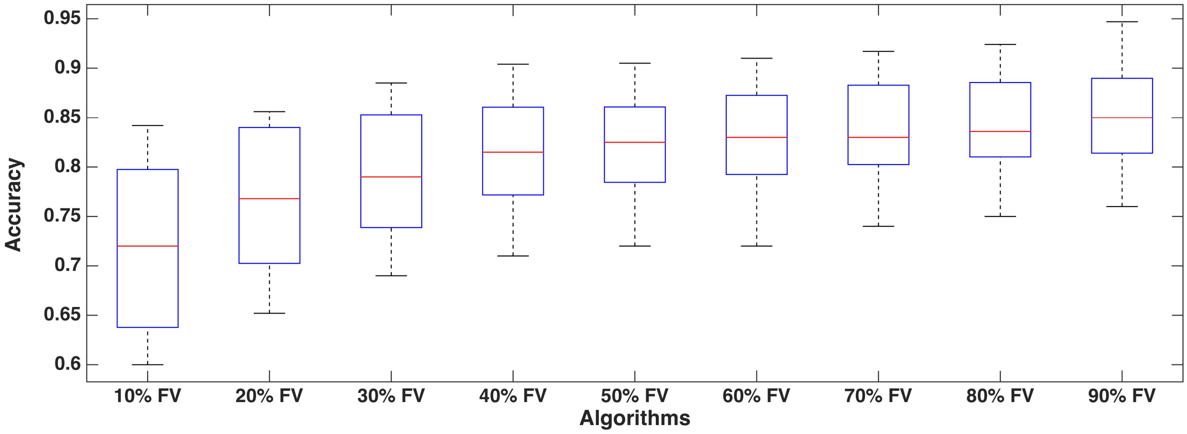


Figure 6.27: Classification accuracy of the FV feature mapping with a K-NN classifier using MFCC features, as we vary the percentage of labeled data.

we vary the percentage of labeled data from 10% to 90% by an increment of 10% and report the classification results of the unlabeled data. This experiment is performed using the proposed FV feature mapping with MFCC, PLP, and LPCC features. The results are reported in Figures 6.27 - 6.29. As it can be seen, the accuracy improves significantly as we increase the percentage of labeled samples from 10% to 40%. However, increasing the percentage of labeled samples beyond 50% provide only a slight improvement in classifying unlabeled samples.

In Figure 6.30, we compare the speaker identification accuracy using standard supervised learning with a K-NN classifier (as reported in Section 6.8.3.2) with semi-supervised learning with 80% labeled samples (i.e. both methods use the same percentage of labeled samples). As it can be seen, one advantage of using the semi-supervised approach is that reasonable results can be obtained using very few labeled samples (10%). This is a desirable feature in speaker identification as the labeling process can be tedious.

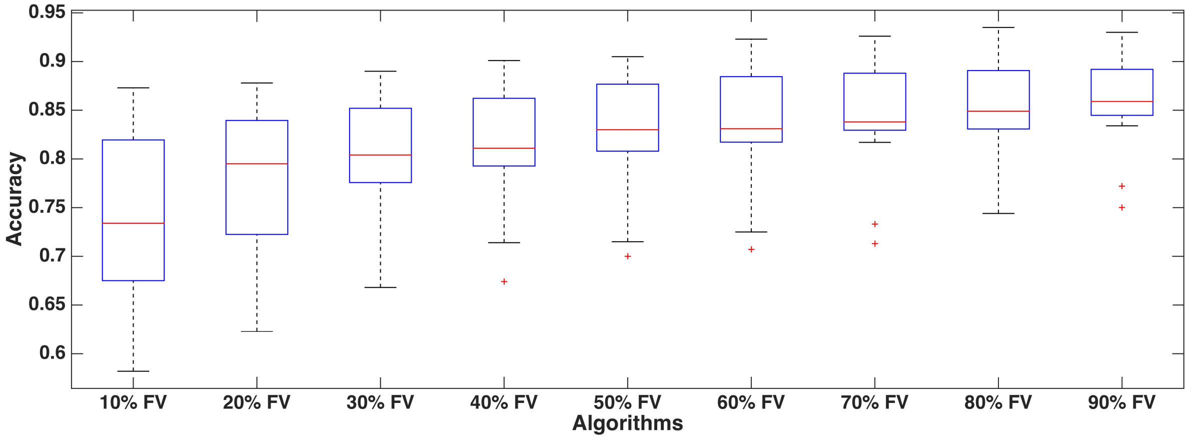


Figure 6.28: Classification accuracy of the FV feature mapping with a K-NN classifier using PLP features, as we vary the percentage of labeled data.

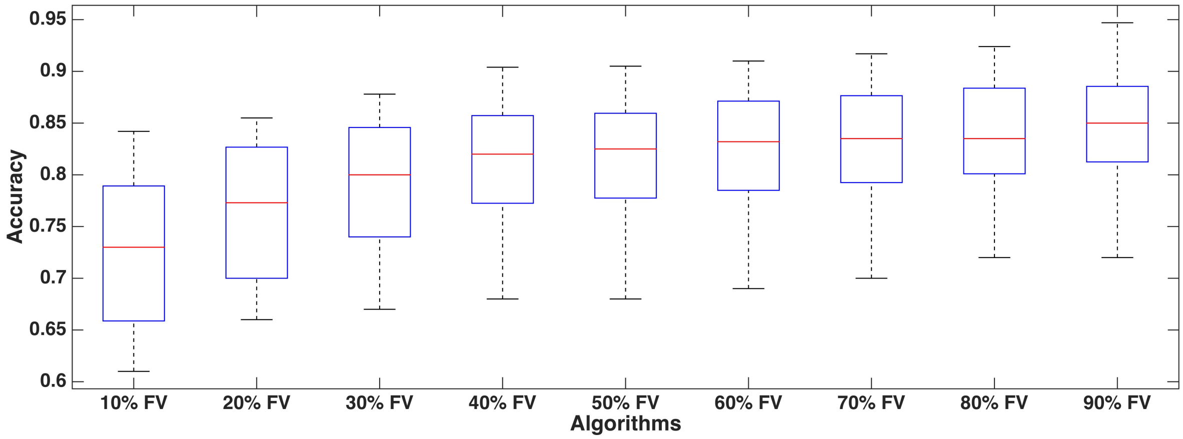


Figure 6.29: Classification accuracy of the FV feature mapping with a K-NN classifier using LPCC features, as we vary the percentage of labeled data.

6.8.3.4 Emotion Recognition

Training a classifier for emotion recognition requires a large collection of labeled training data. Unfortunately, our 15 data sets are not labeled with respect to the speaker's emotion and labeling them is a tedious task. Instead, we use existing public data sets for training, and we test the learned classifiers on our data.

Many databases have been used for audio and/or video based emotion recognition [96]. In our experiments, to evaluate the performance of the proposed soft BoW

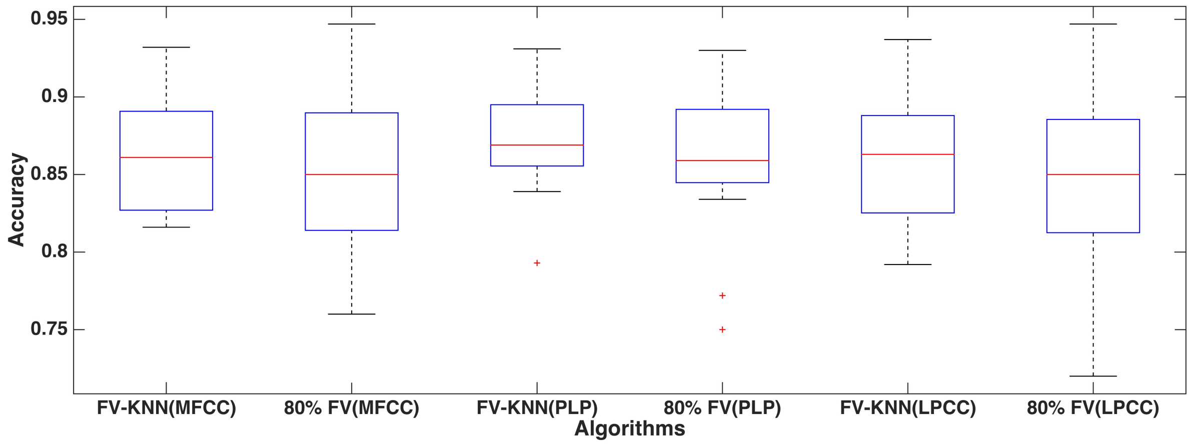


Figure 6.30: Comparison the speaker identification accuracy using standard supervised learning with a K-NN classifier with semi-supervised learning with 80% labeled samples.

TABLE 6.4

EMO-DB description

	Anger	Boredom	Disgust	Fear	Happiness	Sadness	Neutral	Total
Male	60	35	11	36	27	25	39	233
Female	67	46	35	33	44	37	40	302
Total	127	81	46	69	71	62	79	535

and FV feature representation approaches on speech emotion recognition, we use a well-known public free database, named Berlin Emotional database (EMO-DB). It contains about 535 utterances by 10 speakers (5 male and 5 female speakers). Seven emotional states are represented in this data: anger (A), boredom (B), disgust (D), anxiety/fear (F), happiness (H), sadness (S), and neutral (N). A total of 233 utterances were spoken by males, and the remaining 302 utterances were spoken by females. Each utterance is 2 to 4 seconds long. Table 6.4 summarizes the statistics of the EMO-DB database.

First, 12 dimensional low-level features are extracted from each emotional segment in the database. Then, features are mapped to histograms using our BoW and

TABLE 6.5

EMO-DB speakers emotion recognition based on MFCC feature

	Male	Female	Mix
SV-KNN	0.698	0.714	0.697
SV-SVM	0.758	0.766	0.722
C-BoW-NB	0.772	0.793	0.77
F-BoW-NB	0.81	0.823	0.808
P-BoW-NB	0.785	0.803	0.78
FV-KNN	0.822	0.859	0.828

FV methods.

As in speaker identification, we use a NB classifier for the BoW mappings and a K-NN classifier for the FV mapping. We compare the results to those obtained using the GMM mean supervector [127] with K-NN classifier (SV-KNN), and SVM classifier (SV-SVM) [129]. We compare the different methods using 4 different features: MFCC, PLP, LPCC, and GFCC. We report the results using subsets of the data that contain either male or female speakers as well as the results using all speakers. As in earlier experiments, for GMM mean supervector methods, we set the number of Gaussian components to 100. The number of Gaussian components in FV feature is also set 100. For the soft BoW methods, the initial number of prototypes for each emotion class is set to 20.

The results are reported in Tables 6.5 - 6.8. As it can be seen, for all 4 features, the proposed BoW and FV mappings outperform existing methods. The FV features have a slight improvement over the soft BoW features.

TABLE 6.6

EMO-DB speakers emotion recognition based on PLP feature

	Male	Female	Mix
SV-KNN	0.68	0.67	0.661
SV-SVM	0.783	0.776	0.754
C-BoW-NB	0.812	0.828	0.819
F-BoW-NB	0.822	0.846	0.831
P-BoW-NB	0.83	0.852	0.836
FV-KNN	0.831	0.866	0.85

TABLE 6.7

EMO-DB speakers emotion recognition based on LPCC feature

	Male	Female	Mix
SV-KNN	0.767	0.783	0.772
SV-SVM	0.792	0.811	0.803
C-BoW-NB	0.8	0.813	0.806
F-BoW-NB	0.842	0.865	0.84
P-BoW-NB	0.812	0.835	0.818
FV-KNN	0.843	0.872	0.825

TABLE 6.8

EMO-DB speakers emotion recognition based on GFCC feature

	Male	Female	Mix
SV-KNN	0.676	0.683	0.677
SV-SVM	0.711	0.735	0.713
C-BoW-NB	0.702	0.721	0.71
F-BoW-NB	0.74	0.769	0.745
P-BoW-NB	0.72	0.741	0.714
FV-KNN	0.738	0.769	0.72

6.9 Content-based Segmentation and Retrieval of Medical Simulation Video

The proposed components, including speaker segmentation, speaker identification, and emotion recognition algorithms, are integrated within a graphical user interface (GUI) to help the physician review and navigate through the video simulations. Figure 6.31 illustrates the flowchart of the developed GUI to aid the physician review medical simulation video. First, the physician selects one of the simulations from the database and identifies the speakers involved in the simulation. The system then loads the model of each selected speaker (from previous training). Second, the audio stream is extracted from the video, preprocessed, and segmented. Third, a speaker is assigned to each segment using a trained classifier. Finally, the results are presented to the user in an intuitive and interactive format. For each segment, we display its length (in seconds) and the confidence of the speaker's identity, speech emotion, and other relevant information. The physician can select any of segments and play the video clip.

Figure 6.32 shows the initial interface of our GUI. It requires the user to provide 3 input parameters: (1) video session to be processed; (2) File that has the parameters of trained models of all speakers in the database, and (3) File that has all parameters setting. Several key components are designed to display various information in multiple panels. Panel 1 allows the user to play the original loaded video. Panel 2 shows the speakers that have trained models in the database. Panel 3 is used to play a selected video segment that was identified to belong to a given speaker. Panel 4 is used to show the speaker's identification or emotion recognition results. Panel 5 is used to display the number of segments identified by each speaker. Panel 6 is used to display a transcript of the selected video segment, and panel 7 is used for

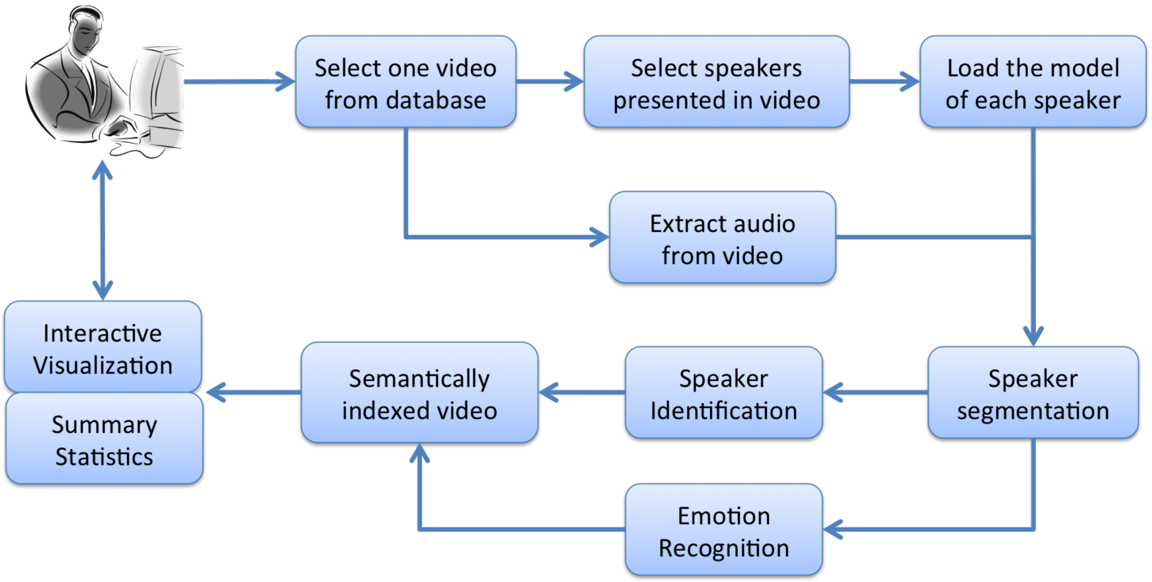

Select one video from database
Select speakers presented in video
Load the model of each speaker
Extract audio from video
Interactive Visualization
Summary Statistics
Semantically indexed video
Speaker Identification
Speaker segmentation
Emotion Recognition

Figure 6.31: Flowchart of developed GUI to aid the physician in reviewing medical simulation data.

the emotion recognition results.

Figure 6.33 shows the parameters settings for various stages of our system. These include audio preprocessing, audio feature selection, feature setting, speaker segmentation algorithms, speaker identification or emotion recognition algorithms. Some of these features and algorithms are those proposed in this dissertation. Others are existing methods used in our comparison and analysis.

Figure 6.34 displays sample results from the speaker identification component. At the bottom of this figure, we display the statistics of this simulation. For instance, in Panel A we display the total time used by each speaker. In Panel B, we show a chronological order of when and how long each person spoke.

Figure 6.35 shows sample results from the emotion recognition component.

Similar to the speaker identification results, the speakers' information for each emotion are provided in panel B, and the emotion recognition result (with probability for every selected segment) is shown in panel A.

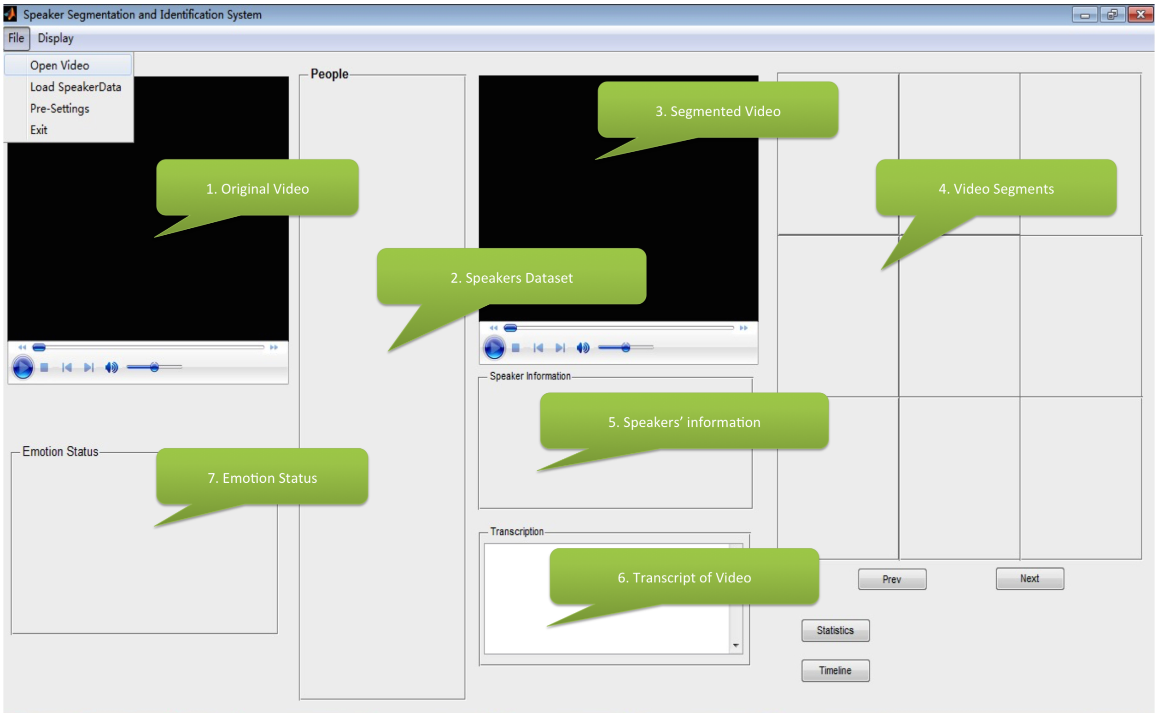
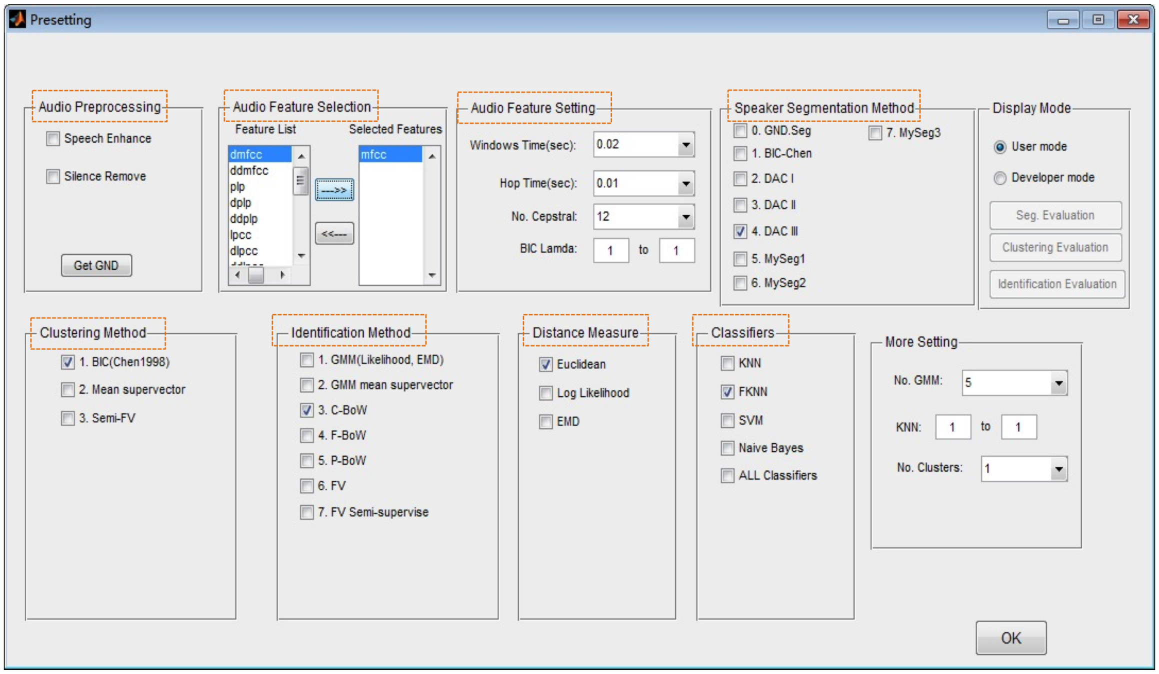


Figure 6.32: Initial interface of the GUI with descriptor of its 7 panels.

Using this simple GUI, the physician can efficiently identify "Who Spoke, When, and what was the emotion". This is very important because the physician needs to review these video simulations on a regular basis. This segmentation and visualization system can also generate simple, but very useful statistics that summarize the entire simulation session in a completely unsupervised way. For instance, it can provide the percentage of time during which each speaker spoke. Typically, it is expected that the resident/nurse uses less time than the patient. The proposed interface could also be used to identify segments where the patient was interrupted, tone of voice, etc.



Presetting

Audio Preprocessing
- ☐ Speech Enhance
- ☐ Silence Remove

Audio Feature Selection
- Feature List: dmfcc, ddmfcc, plp, dplp, ddplp, lpcc, dlpcc
- Selected Features: mfcc

Audio Feature Setting
- Windows Time(sec): 0.02
- Hop Time(sec): 0.01
- No. Cepstral: 12
- BIC Lamda: 1 to 1

Speaker Segmentation Method
- ☐ 0. GND.Seg
- ☐ 1. BIC-Chen
- ☐ 2. DAC I
- ☐ 3. DAC II
- ☑ 4. DAC III
- ☐ 5. MySeg1
- ☐ 6. MySeg2
- ☐ 7. MySeg3

Display Mode
- ● User mode
- ○ Developer mode

Clustering Method
- ☑ 1. BIC(Chen1998)
- ☐ 2. Mean supervector
- ☐ 3. Semi-FV

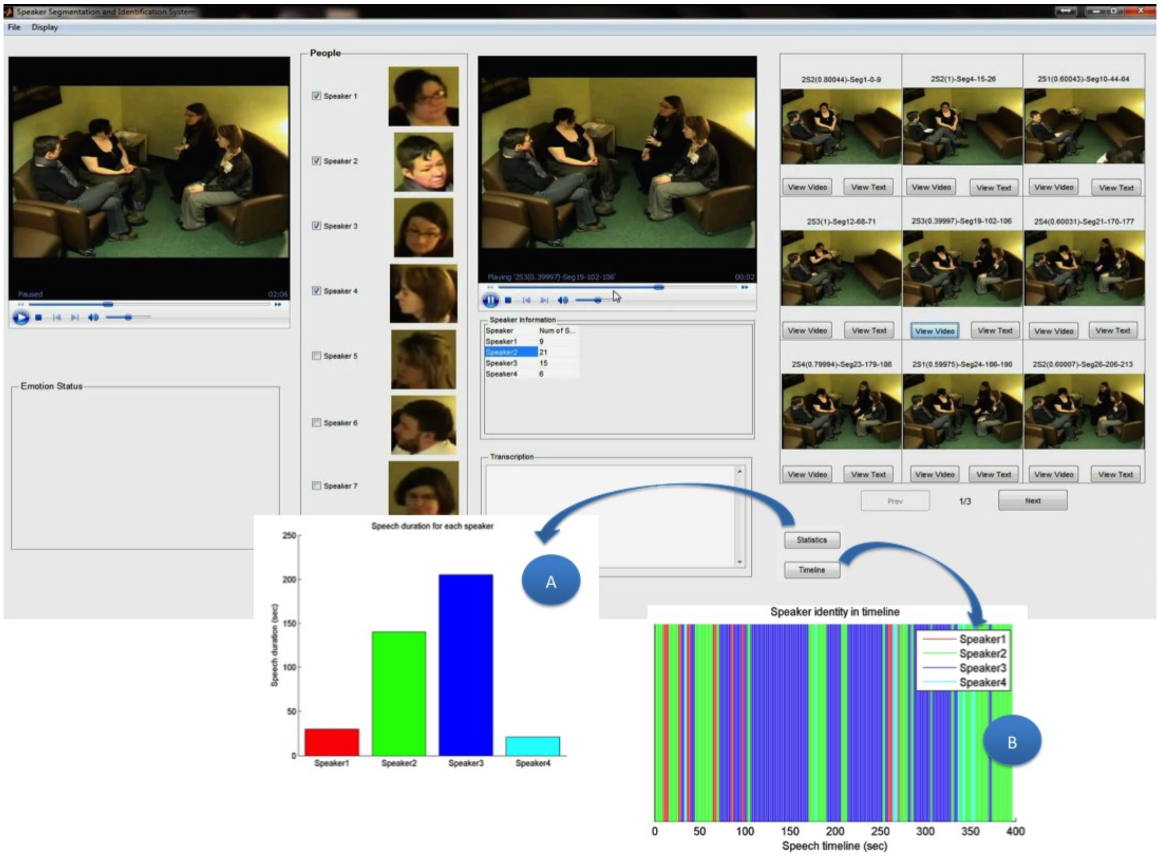
Identification Method
- ☐ 1. GMM(Likelihood, EMD)
- ☐ 2. GMM mean supervector
- ☑ 3. C-BoW
- ☐ 4. F-BoW
- ☐ 5. P-BoW
- ☐ 6. FV
- ☐ 7. FV Semi-supervise

Distance Measure
- ☑ Euclidean
- ☐ Log Likelihood
- ☐ EMD

Classifiers
- ☐ KNN
- ☑ FKNN
- ☐ SVM
- ☐ Naive Bayes
- ☐ ALL Classifiers

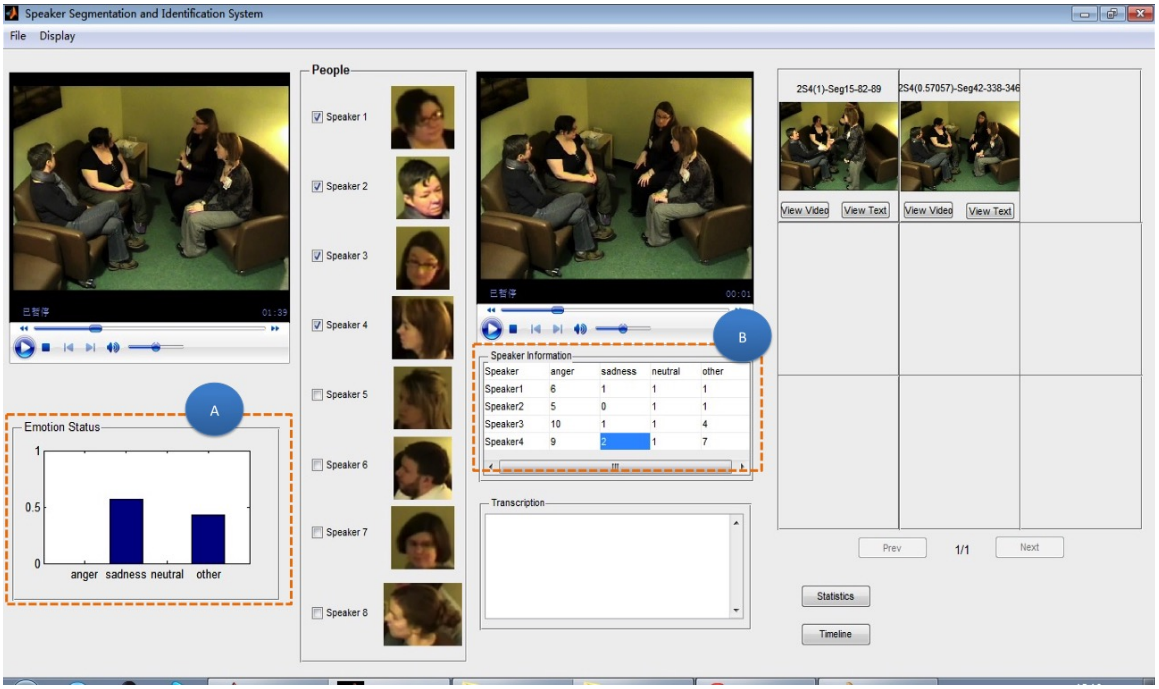
More Setting
- No. GMM: 5
- KNN: 1 to 1
- No. Clusters: 1

Figure 6.33: Various parameters that the user can modify. Each parameter has a default value that was optimized in our experiment.



Speaker	Num of S...
Speaker1	9
Speaker2	21
Speaker3	15
Speaker4	6

Figure 6.34: Visualization of the speaker identification results.



Speaker Information

Speaker	anger	sadness	neutral	other
Speaker1	6	1	1	1
Speaker2	5	0	1	1
Speaker3	10	1	1	4
Speaker4	9	2	1	7

Figure 6.35: Visualization of the emotion recognition results.

CHAPTER 7

CONCLUSIONS AND POTENTIAL FUTURE WORK

7.1 Conclusions

We have developed and implemented methods for the extraction, integration, analysis, and presentation of knowledge from video recordings of medical simulations. Our goal was to provide the physicians with tools to efficiently retrieve video shots that relate to "**who spoke, when, and what was the emotion of the speaker**".

Three main area were researched: speaker segmentation, speaker identification, and emotion recognition. The objective of the speaker segmentation is to detect speaker change boundaries in an audio stream and segment the corresponding video into shots, where only one speaker should be included within each shot. Speaker segmentation provides a fundamental preprocessing step for speaker identification and emotion recognition. In our approach, first, the audio component is extracted from the video recording. Then, various low-level audio features are extracted to detect and remove silence segments. We implemented, tested, and compared two different approaches for this task. The remaining speech (non-silent) segments are analyzed further to identify speaker changing points and locate the corresponding video shots. For this speaker segmentation task, we proposed two methods that can fuse the intermediate results of multiple segmentation algorithms. These are the extrema point-level fusion, and the distance level fusion algorithms. We compared our proposed methods with five different speaker segmentation algorithms: Bayesian

Information Criteria (BIC) segmentation [53], sequential BIC [70], and three Divide-and-Conquer (DAC) based methods [71]. We showed that our methods can detect more true speaker changing points resulting in more pure segments for further processing.

In speaker identification, each segment is classified into a predefined class. For this component, we proposed two feature representation methods: soft bag-of-words (BoW) mapping and Fisher Vector (FV) mapping. BoW feature mapping transforms low-level audio streams to more meaningful feature descriptors using two main steps: (1) clustering of low-level speech features and prototype generation, and (2) membership mapping (crisp, fuzzy, or possibilistic) and histogram-based feature construction. FV feature mapping is a generalization of the BoW feature representation. It uses the Fisher Kernel principle and combines the benefits of generative and discriminative approaches by computing the gradient of the sample log-likelihood with respect to the model parameters. The main advantage of the proposed BoW and FV mappings is that speech segments of different lengths are mapped to feature vectors of equal dimensions. Thus, standard classifiers could be used for this task. Using 15 simulation sessions, we showed that our feature mappings, coupled with standard classifiers, outperform state-of-the-art algorithms for both speaker identification and emotion recognition.

Data labeling, for the purpose of classifiers' training, is a tedious and time consuming task. Thus, if an additional speaker is added to the simulation database, a considerable amount of time would be needed to collect and label speech segments for this speaker. An alternative approach is to use semi-supervised learning where only a limited amount of labeled data is needed and the learning algorithm will label the remaining data based on its proximity to the labeled data. In our developed system, we used the proposed FV feature representation and adapted a learning algorithm that is based on label propagation. We showed that this semi-supervised approach can perform as good as a completely supervised approach using only 30% to 40% of the labeled data.

We have integrated the above components and developed a GUI prototype that processes medical simulation video and allows doctors to browse videos and retrieve shots that identify ”who spoke, when, and what was the emotion of the speaker”. The GUI prototype also generates summary statistics such as: for how long did each person spoke? What is the longest uninterrupted speech segment? Is there an unusual large number of pauses within the speech segment of a given speaker?

The performance of the proposed system can be improved by upgrading and adding sensors to the current data collection system. For instance, by making each speaker wear a microphone, the quality of the audio can improve significantly. In addition to improving the recognition rates, improved audio quality may make it possible to transcribe the speech segments. Similarly, using a camera with higher resolution will make it possible to use visual cues such as facial expressions, or when one of the speakers leaves/enters the room.

7.2 Potential Future Work

In our developed system, the silence during the video simulation has been detected and removed. In fact, the silence can be a good feature. For example, the silence happened during two speakers may have difference meanings from the silence happened within one speaker's talk. This may indicate the change of the speakers' emotions during the conversation or the communication between two speakers. In the future work, the attribution of the silence may be further investigated.

The Roter Interaction Analysis System (RIAS) [132–134] has been developed to analyse doctor-patient communication during conventional face-to-face consultations. The RIAS is used to quantify communication events, which may be correlated with patient, provider, and system attributes and health outcomes. Our proposed features can be fit to RIAS system for the interaction analysis.

REFERENCES

[1] S. Meignier, D. Moraru, C. Fredouille, J. Bonastre, and L. Besacier, “Step-by-step and integrated approaches in broadcast news speaker diarization,” *Computer Speech and Language*, vol. 20, no. 2-3, pp. 303–330, April 2006.

[2] Y. Wang, Z. Liu, and J. Huang, “Multimedia content analysis using both audio and visual clues,” *IEEE Signal Processing Magazine*, 2000.

[3] T. Zhang and C. Kuo, “Audio content analysis for online audiovisual data segmentation and classification,” *IEEE Trans. Speech and Audio Processing*, vol. 9, no. 4, pp. 441–451, May 2001.

[4] V. Apsingekar and P. Leon, “Speaker model clustering for efficient speaker identification in large population applications,” *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 17, no. 4, pp. 848–853, 2009.

[5] D. Vijayasenan, F. Valente, and H. Bourlard, “An information theoretic approach to speaker diarization of meeting data,” *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 17, no. 7, pp. 1382–1393, 2009.

[6] C. Hsu and L. Lee, “Higher order cepstral moment normalization for improved robust speech recognition,” *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 17, no. 2, pp. 205–220, 2009.

[7] D. Reynolds, T. Quatieri, and R. Dunn, “Speaker verification using adapted gaussian mixture models,” *Digital Signal Processing*, vol. 10, pp. 19–41, 2000.

[8] M. Ferras, C. Leung, C. Barras, and J. Gauvain, “Comparison of speaker adaptation methods as feature extraction for svm-based speaker recognition,” *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 18, no. 5, pp. 1366–1378, 2010.

[9] A. Hauptmann, “Automatic spoken document retrieval,” *Computer Science Department*, 2006.

[10] G. Guo and S. Li, “Content-based audio classification and retrieval by support vector machines,” *IEEE Transactions on Neural Networks*, vol. 14, no. 1, pp. 209–215, 2003.

[11] S. Kiranyaz and M. Gabbouj, “Generic content-based audio indexing and retrieval framework,” *IEE Proc. Vis. Image Signal Process.*, vol. 153, no. 3, pp. 285–297, 2006.

[12] C. Barras, X. Zhu, S. Meignier, and J. Gauvain, “Multistage speaker diarization of broadcast news,” *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 14, no. 5, pp. 1505–1512, 2006.

[13] T. Stafylakis, V. Katsouros, and G. Carayannis, "The segmental bayesian information criterion and its applications to speaker diarization," *IEEE Journal of Selected Topics in Signal Processing*, vol. 4, no. 5, pp. 857–886, 2010.

[14] N. Bassiou, V. Moschou, and C. Kotropoulos, "Speaker diarization exploiting the eigengap criterion and cluster ensembles," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 18, no. 8, pp. 2134–2144, 2010.

[15] S. Guerlain, B. Turrentine, R. Adams, and J. Calland, "Using video data for the analysis and training of medical personnel," *Cogn. Tech. Work*, vol. 6, pp. 131–138, 2004.

[16] A. Jain, A. Ross, and S. Prabhakar, "An introduction to biometric recognition," *IEEE Trans. Circuits and Systems for Video Technology*, vol. 14, no. 1, pp. 4–20, 2004.

[17] A. Jain, K. Nandakumar, and A. Ross, "Score normalization in multimodal biometric systems," *Pattern Recognition*, vol. 38, pp. 2270–2285, 2005.

[18] M. Gales, D. Kim, P. Woodland, H. Chan, D. Mrva, R. Sinha, and S. Tranter, "Progress in the cu-htk broadcast news transcription system," *IEEE Trans. Speech and Audio Processing*, vol. 14, no. 5, pp. 1513–1525, September 2006.

[19] K. Lee and D. Ellis, "Audio-based semantic concept classification for consumer video," *IEEE Trans. Speech and Audio Processing*, vol. 18, no. 6, pp. 1406–1416, August 2010.

[20] X. Anguera and J. Pardo, "Robust speaker diarization for meetings: Icsi rt06s evaluation system," in *in Proc. ICSLP*, 2006.

[21] C. Jung, M. Kim, and H. Kang, "Selecting feature frames for automatic speaker recognition using mutual information," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 18, no. 6, pp. 1332–1340, 2010.

[22] D. Zhu, B. Ma, and H. Li, "Speaker verification with feature-space maplr parameters," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 19, no. 3, pp. 1332–1340, 2011.

[23] S. Li, "Content-based audio classification and retrieval using the nearest feature line method," *IEEE Transactions on Speech and Audio Processing*, vol. 8, no. 5, pp. 619–625, 2000.

[24] S. Kiranyaz, A. Qureshi, and M. Gabbouj, "A generic audio classification and segmentation approach for multimedia indexing and retrieval," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 14, no. 3, pp. 1062–1081, 2006.

[25] C. Chelba, T. Hazen, and M. Saraclar, "Retrieval and browsing of spoken content," *IEEE Signal Processing Magazine*, pp. 39–49, 2008.

[26] J. Hansen, R. Huang, B. Zhou, M. Seadle, J. Deller, A. Gurijala, M. Kurimo, and P. Angkititrakul, "Speechfind: advances in spoken document retrieval for a national gallery of the spoken word," *IEEE Transactions on Speech and Audio Processing*, vol. 13, no. 5, pp. 712–730, 2005.

[27] C. Huang and C. Wu, "Spoken document retrieval using multilevel knowledge and semantic verification," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 15, no. 8, pp. 2551–2560, 2007.

[28] K. Park, J. Park, and Y. Oh, "Gmm adaptation based online speaker segmentation for spoken document retrieval," *IEEE Transactions on Consumer Electronics*, vol. 56, no. 2, pp. 1123–1129, 2010.

[29] M. Kotti, D. Ververidis, G. Evangelopoulos, I. Panagakis, C. Kotropoulos, P. Maragos, and I. Pitas, "Audio-assisted movie dialogue detection," *IEEE Trans. Circuits and Systems for Video Tech.*, vol. 18, no. 11, pp. 1618–1627, 2008.

[30] D. Wyatt, T. Choudhury, and H. Kautz, "Capturing spontaneous conversation and social dynamics: A privacy sensitive data collection effort," in *in Proc. of ICASSP*, 2007.

[31] S. Jing, B. Kane, and S. Luz, "Automatic content segmentation of audio recordings at multidisciplinary medical team meetings," in *1st International Conf. on Information Technology*, 2008, pp. 1–4.

[32] B. Pardo, "Finding structure in audio for music information retrieval," *IEEE Signal Processing Magazine*, pp. 126–132, 2006.

[33] A. Ross and A. Jain, "Multimodal biometrics: an overview," in *in Proc. 12th Euro. Signal Process. Conf.*, 2004, pp. 1221–1224.

[34] "http://www.icsi.berkeley.edu/speech/docs/htkbook3.2/," .

[35] S. Quackenbush and A. Lindsay, "Overview of mpeg-7 audio," *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 11, no. 6, pp. 725–729, June 2001.

[36] S. S. Stevens, J. Volkmann, and E. B. Newman, "A scale for the measurement of the psychological magnitude pitch," *The Journal of the Acoustical Society of America*, vol. 8, no. 3, pp. 155–210, 1937.

[37] C. You, K. Lee, and H. Li, "Gmm-svm kernel with a bhattacharyya-based distance for speaker recognition," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 18, no. 6, pp. 1300–1312, 2010.

[38] Q. Li and Y. Huang, "Robust speaker identification using an auditory-based feature," in *ICASSP*, 2010.

[39] D. Reynolds, "Speaker identification and verification using gaussian mixture models," *Speech Commun.*, vol. 17, pp. 91–105, 1995.

[40] J. Makhoul, F. Kubala, T. Leek, D. Liu, L. Nguyen, R. Schwartz, and A. Srivastava, "Speech and language technologies for audio indexing and retrieval," *PROCEEDINGS OF THE IEEE*, vol. 88, no. 8, pp. 1338–1353, 2000.

[41] H. Hermansky, "Perceptual linear predictive (plp) analysis of speech," *J. Acoust. Soc. Am.*, vol. 87, no. 4, pp. 1738–1752, 1990.

[42] H. Hermansky, N. Morgan, A. Bayya, and P. Kohn, “Rasta-plp speech analysis technique,” in *in ICASSP*, 1992, pp. 121–124.

[43] P. Delsarte and Y. Genin, “The split levinson algorithm,” *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 34, no. 3, pp. 470–478, 1986.

[44] M. Kotti, E. Beneto, and C. Kotropoulos, “Computationally efficient and robust bic-based speaker segmentation,” *IEEE Trans. Audio, Speech, and Language processing*, vol. 16, no. 5, pp. 920–933, 2008.

[45] L. Rabiner and R. Schafer, *Digital Processing of Speech Signals*, Englewood Cliffs, New Jersey, Prentice Hall, 1978.

[46] J. Grey and J. Gordon, “Perceptual effects of spectral modifications on musical timbres,” *Journal of the Acoustical Society of America*, vol. 63, no. 5, pp. 1493–1500, 1978.

[47] C. Wu and C. Hsieh, “Multiple change-point audio segmentation and classification using an mdl-based gaussian model,” *IEEE Trans. Audio, Speech, and Language processing*, vol. 14, no. 2, pp. 647–657, 2006.

[48] F. Kubala and et al., “The 1996 bbn byblos hub-4 transcription system,” *Proceedings of the Speech Recognition Workshop*, pp. 90–93, 1997.

[49] R. Bakis and et al., “Transcription of bn shows with the ibm lvcsr system,” *in Proc. DARPA Speech Recognition Workshop*, 1997.

[50] H. Beigi and S. Maes, “Speaker, channel and environment change detection,” *in Proc. World Congr. Automation*, 1998.

[51] H. Gish and N. Schmidt, “Text-independent speaker identification,” *IEEE Signal Processing Mag.*, pp. 18–21, 1994.

[52] M. Siegler, U. Jain, B. Raj, and R. Stern, “Automatic segmentation, classification and clustering of broadcast news audio,” *in Proc. DARPA Speech Recognition Workshop*, pp. 97–99, 1997.

[53] S. S. Chen and P. S. Gopalakrishnan, “Speaker, environment and channel change detection and clustering via the bayesian information criterion,” in *in Proc. DARPA Broadcast News Transcription Understanding Workshop*, Landsdowne, VA, 1998.

[54] Cettolo M and M. Vescovi, “Efficient audio segmentation algorithms based on the bic,” in *in Proc. ICASSP*, 2003.

[55] S. Know and S. Narayanan, “Unsupervised speaker indexing using generic models,” *IEEE Trans. Speech Audio Process.*, vol. 13, no. 5, pp. 1004–1013, 2005.

[56] M. Collet, D. Charlet, and F. Bimbot, “A correlation metric for speaker tracking using anchor models,” in *in ICASSP*, 2005.

[57] H. Kim, D. Elter, and T. Sikora, “Hybrid speaker-based segmentation system using model-level clustering,” in *in ICASSP*, 2005.

[58] A. Tritschler and R. Gopinath, "Improved speaker segmentation and segments clustering using the bayesian information criterion," in *in Proc. 6th Eur. Conf. Speech Commun. Techol.*, 1999, pp. 679–682.

[59] J. Arias, J. Pinquier, and R. AndreObrecht, "Evaluation of classification techniques for audio indexing," in *in Proc. 13th Eur. Signal Process. Conf.*, 2005.

[60] J. Hung, H. Wang, and L. Lee, "Automatic metric-based speech segmentation for broadcast news via principal component analysis," in *in Proc. Int. Conf. Spoken Lang. Process.*, 2000, pp. 121–124.

[61] S. Know and S. Narayanan, "Speaker change detection using a new weighted distance measure," in *in Proc. Int. Conf. Spoken Lang.*, 2002, vol. 4, pp. 2537–2540.

[62] B. Zhou and J. Hansen, "Efficient audio stream segmentation via the combined t2 statistic and bayesian information criteria," *IEEE Trans. Speech Audio Process.*, vol. 13, no. 4, pp. 467–474, 2005.

[63] P. Sivakumaran, A. M. Ariyaeeinia, and J. Fortuna, "An effective unsupervised scheme for multiple-speaker-change detection," in *in Proc. Int. Conf. Spoken Lang. Process.*, 2002, pp. 569–572.

[64] L. Lu and H. Zhang, "Speaker change detection and tracking in real-time news broadcasting analysis," in *in Proc. ACM Multimedia*, 2002, pp. 602–610.

[65] A. S. Malegaonkar, A. M . Ariyaeeinia, and P. Sivakumaran, "Efficient speaker change detection using adapted gaussian mixture models," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 15, no. 6, pp. 1859–1869, 2007.

[66] P. C. Lin, J. C. Wang, J. F. Wang, and H. C. Sung, "Unsupervised speaker change detection using svm training misclassification rate," *IEEE Trans. Comput.*, vol. 56, no. 9, pp. 1212–1223, 2007.

[67] P. Delacourt and C. J. Wellekens, "Distbic: A speaker-based segmentation for audio data indexing," *Speech Commun.*, vol. 32, no. 1, pp. 111–126, 2000.

[68] X. Anguera, "Xbic: Real-time cross probabilities measure for speaker segmentation," Tech. Rep., ICSI Technical Report TR-05-008, 2005.

[69] S. Cheng and H. Wang, "Metric seqdac: A hybrid approach for audio segmentation," in *in Proc. 8th int. Conf. Spoken Lang. Process.*, 2004, pp. 1617–1620.

[70] S. Cheng and H. Wang, "A sequential metric-based audio segmentation method via the bayesian information criterion," in *In Proceedings of INTERSPEECH*, 2003.

[71] S. Cheng, H. Wang, and H. Fu, "Bic-based speaker segmentation using divide-and-conquer strategies with application to speaker diarization," in *IEEE Trans. on Audio, Speech, and Language Processing*, 2010, vol. 18 of *1*, pp. 141–157.

[72] A. Dempster, N. Laird, and D. Rubin, "Maximum likelihood from incomplete data via the em algorithm," *J. of Royal Stat. Society*, vol. 39, no. 1, pp. 1–38, 1977.

[73] Q. Jin, *Robust speaker recognition*, Ph.d. thesis, 2007.

[74] S. Chu, H. Tang, and T. Huang, "Fishervoice and semi-supervised speaker clustering," in *ICASSP*, 2009.

[75] Y. Rubner, C. Tomasi, and L. Guibas, "The earth mover's distance as a metric for image retrieval," *International Journal of Computer Vision*, vol. 40, no. 2, pp. 99–121, 2000.

[76] H. Ling and K. Okada, "Diffusion distance for histogram comparison," in *CVPR*, 2006, pp. 246–253.

[77] J. Campbell, "Speaker recognition: a tutorial," *Proceedings of the IEEE*, vol. 85, no. 9, pp. 1437–1462, 1997.

[78] M. Siu, X. Yang, and H. Gish, "Discriminatively trained gmms for language classification using boosting methods," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 17, no. 1, pp. 187–197, 2009.

[79] T. Jaakkola and D. Haussler, "Exploiting generative models in discriminative classifiers," in *in Proc. NIPS*, 1999, pp. 487–493.

[80] W. Campbell, D. Sturim, D. Reynolds, and A. Solomonoff, "Svm based speaker verification using a gmm supervector kernel and nap variability compensation," in *in Proc. ICASSP*, 2006, pp. 97–100.

[81] A. Stolcke, L. Ferrer, S. Kajarekar, E. Shriberg, and A. Venkataramam, "Mllr transforms as features in speaker recognition," in *in Interspeech*, 2005, pp. 2425–2428.

[82] H. Yang, Y. Dong, X. Zhao, L. Lu, and H. Wang, "Cluster adaptive training weights as features in svm-based speaker verification," in *in Interspeech*, 2007, pp. 2013–2016.

[83] S. Pigeon, P. Druyts, and P. Verlinde, "Applying logistic regression to the fusion of the nist99 1-speaker submisssions," *Digital Signal Processing*, pp. 237–248, 2000.

[84] C. Longworth and M. Gales, "Combining derivative and parametric kernels for speaker verification," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 17, no. 4, pp. 748–757, 2009.

[85] T. Kinnunen, J. Saastamoinen, V. Hautamaki, M. Vinni, and P. Franti, "Comparing maximum a posteriori vector quantization and gaussian mixture models in speaker verification," in *ICASSP*, 2009.

[86] L. Wang, K. Minami, K. Yamamoto, and S. Nakagawa, "Speaker identification by combining mfcc and phase information in noisy environment," in *ICASSP*, 2010.

[87] A. McCallum and K. Nigam, "A comparison of event models for naive bayes text classification," *AAAI-98 workshop on learning for text categorization*, pp. 41–48, 1998.

[88] G. Csurka, C. Dance, L.X. Fan, J. Willamowski, and C. Bray, "Visual categorization with bags of keypoints," *Proc. of ECCV International Workshop on Statistical Learning in Computer Vision*, 2004.

[89] J. Sivic, "Efficient visual search of videos cast as text retrieval," *IEEE Trans. on pattern analysis and machine intelligence*, vol. 31, no. 4, pp. 591–605, 2009.

[90] K. Boakye B. Peskin, "Text-constrained speaker recognition on a text-independent task," *In ODYS-2004*, pp. 129–134, 2004.

[91] K. Ishiguro, T. Yamada, S. Araki, T. Nakatani, and H. Sawada, "Probabilistic speaker diarization with bag-of-words representations of speaker angle information," *IEEE Trans. on audio, speech, and language processing*, vol. 20, no. 2, pp. 447–460, 2012.

[92] J. Sanchez, F. Perronnin, T. Mensink, and J. Verbeek, "Image classification with the fisher vector: Theory and practice," *International Journal of Computer Vision*, vol. 105, no. 3, pp. 222–245, 2013.

[93] D. Oneata, J. Verbeek, and C. Schmid, "Action and event recognition with fisher vectors on a compact feature set," in *Computer Vision (ICCV), 2013 IEEE International Conference on*, Dec 2013, pp. 1817–1824.

[94] B. Safadi and G. Quenot, "Descriptor optimization for multimedia indexing and retrieval," in *Content-Based Multimedia Indexing (CBMI), 2013 11th International Workshop on*, June 2013, pp. 65–71.

[95] B. Zhou, A. Lapedriza, J. Xiao, A. Torralba, and A. Oliva, "Learning deep features for scene recognition using places database," in *Advances in Neural Information Processing Systems*, 2014.

[96] Z. Zeng, "A survey of affect recognition methods: audio, visual, and spontaneous expressions," *IEEE PAMI*, 2009.

[97] H. Lo, J. Wang, H. Wang, and S. Lin, "Cost-sensitive multilabel learning for audio tag annotation and retrieval," *IEEE TRANSACTIONS ON MULTIMEDIA*, vol. 13, no. 3, pp. 518–529, 2011.

[98] Y. Pan, H. Lee, and L. Lee, "Interactive spoken document retrieval with suggested key terms ranked by a markov decision process," *IEEE TRANSACTIONS ON AUDIO, SPEECH, AND LANGUAGE PROCESSING*, vol. 20, no. 2, pp. 632–645, 2012.

[99] R. Zheng and B. Xu S. Zhang, "Text-independent speaker identification using gmm-ubm and frame level likelihood normalization," in *International Symposium on Chinese Spoken Language Processing*, Dec. 2004, pp. 289–292.

[100] James C. Bezdek, *Pattern Recognition with Fuzzy Objective Function Algorithms*, Kluwer Academic Publishers, Norwell, MA, USA, 1981.

[101] H. Frigui, "Membershipmap: Data transformation based on granulation and fuzzy membership aggregation," *IEEE Trans. Fuzzy Systems*, vol. 14, no. 6, pp. 885–896, Dec 2006.

[102] A. Banerjee and R. Dave, "Validating clusters using the hopkins statistic," Budapest, Hungary, July 2004, FUZZ-IEEE, pp. 25–29.

[103] R.O. Duda, P.E. Hart, and D.G. Stork, "Pattern classification, 2nd edition," *New York: John Wiley & Sons*, 2000.

[104] L. Kaufman and P. Rousseeuw, "Finding groups in data: An introduction to cluster analysis," *New York: Wiley*, 1990.

[105] F. R. Hampel, E. M. Ronchetti, P. J. Rousseeuw, and W. A. Stahel, "Robust statistics the approach based on influence functions," *New York: Wiley*, 1986.

[106] J. Krapac, J. Verbeek, and F. Jurie, "Modeling spatial layout with fisher vectors for image categorization," *ICCV*, 2011.

[107] T. Cover and P. E. Hart, "Nearest neighbor pattern classification," *IEEE Trans. Inf. Theory*, vol. 13, no. 1, pp. 21–27, Jan. 1967.

[108] J. M. Keller, M. R. Gray, and J. A. Givens, "A fuzzy k-nearest neighbor algorithm," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 15, no. 4, pp. 580–585, 1985.

[109] Corinna Cortes and Vladimir Vapnik, "Support-vector networks," *Machine Learning*, vol. 20, no. 3, pp. 273–297, 1995.

[110] TS Furey, N. Cristianini, N Duffy, DW Bednarski, M. Chummer, and D. Haussler, "Support vector machine classification and validation of cancer tissue samples using microarray expression data," *Bioinformatics*, vol. 16, no. 10, pp. 906–914, Oct. 2000.

[111] Ovidiu Ivanciuc, "Applications of support vector machines in chemistry," *Reviews in Computational Chemistry*, pp. 291–400, 2007.

[112] Thorsten Joachims, "Text categorization with support vector machines: learning with many relevant features," *ECML*, 1998.

[113] Chih-Wei Hsu and Chih-Jen Lin, "A comparison of methods for multiclass support vector machines," *Neural Networks, IEEE Transactions on*, vol. 13, no. 2, pp. 415–425, Mar 2002.

[114] J. C. Platt, N. Cristianini, and J. Shawe-Taylor, "Large margin dags for multiclass classification," in *Advances in Neural Information Processing Systems*, 2000, pp. 547–553.

[115] X. Zhu, "Semi-supervised learning literature survey," Tech. Rep., Computer Science TR 1530, University of Wisconsin-Madison, July 2008.

[116] Xiaojin Zhu and Zoubin Ghahramani, "Learning from labeled and unlabeled data with label propagation," Tech. Rep., 2002.

[117] Dengyong Zhou, Olivier Bousquet, Thomas Navin Lal, Jason Weston, and Bernhard Scholkopf, "Learning with local and global consistency," *Advances in Neural Information Processing Systems 16*, pp. 321–328, 2004.

[118] Xiaojin Zhu, Jaz Kandola, Zoubin Ghahramani, and John Lafferty, "Nonparametric transforms of graph kernels for semi-supervised learning," *Advances in Neural Information Processing Systems*, 2005.

[119] Xiaojin Zhu, Zoubin Ghahramani, and John Lafferty, "semi-supervised learning using gaussian fields and harmonic functions," *ICML*, pp. 912–919, 2003.

[120] T. Gerkmann and R. Hendriks, "Unbiased mmse-based noise power estimation with low complexity and low tracking delay," *IEEE Trans Audio, Speech, Language Processing*, vol. 20, no. 4, pp. 1383–1393, May 2012.

[121] N. Cristianini and J. Shawe-Taylor, *An Introduction to Support Vector Machines and Other Kernel-based Learning Methods*, Cambridge University Press, first edition, 2000.

[122] D. Malah, "Time-domain algorithms for harmonic bandwidth reduction and time scaling of speech signals," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 27, no. 2, pp. 121–133, April 1979.

[123] T. Giannakopoulos, A. Pikrakis, and S. Theodoridis, "A multi-class audio classification method with respect to violent content in movies using bayesian networks," IEEE 9th workshop on Multimedia Signal Processing, October 2007, pp. 90–93.

[124] X. Huang, A. Acero, and H. Hon, "Spoken language processing: a guide to theory, algorithm, and system development," *Prentice-Hall, New Jersey*, 2001.

[125] Q. Wu, L. Zhang, and G. Shi, "Robust feature extraction for speaker recognition based on constrained nonnegative tensor," *J. Comput. Sci. Technol.*, vol. 25, no. 4, pp. 745–754, 2010.

[126] Kshitiz Kumar, Chanwoo Kim, and Richard M. Stern, "Delta-spectral cepstral coefficients for robust speech recognition," *ICASSP*, pp. 4784–4787, 2011.

[127] Tomi Kinnunen and Haizhou Li, "An overview of text-independent speaker recognition: from features to supervectors," *Speech Communication*, vol. 52, no. 1, pp. 12–40, Jan. 2010.

[128] K. J. Han and S. S. Narayanan, "Agglomerative hierarchical speaker clustering using incremental gaussian mixture cluster modeling," in *in Proceedings of Interspeech 2008*, Brisbane, Australia, September 2008.

[129] H. Hu, "Gmm supervector based svm with spectral features for speech emotion recognition," *ICASSP*, 2007.

[130] J. Ajmera, I. McCowan, and H. Bourlard, "Robust speaker change detection," *IEEE Signal Processing Letters*, vol. 11, no. 8, pp. 649–651, 2004.

[131] C. M. Bishop, "Pattern recognition and machine learning," *Springer*, 2006.

[132] Ong LM, Visser MR, Kruyver IP, Bensing JM, van den Brink-Muinen A, Stouthard JM, Lammes FB, and de Haes JC, "The roter interaction analysis system (rias) in oncological consultations: psychometric properties," *Psychooncology*, vol. 7, no. 5, pp. 387–401, 1998.

[133] Roter D and Larson S, "The roter interaction analysis system (rias): utility and flexibility for analysis of medical interactions," *Patient Educ Couns*, vol. 46, no. 4, pp. 243–251, 2002.

[134] Miller EA and Nelson EL, "Modifying the roter interaction analysis system to study provider-patient communication in telemedicine: promises, pitfalls, insights, and recommendations," *Telemed J E Health*, vol. 11, no. 1, pp. 44–55, 2005.

CURRICULUM VITAE

NAME: Shuangshuang Jiang

ADDRESS:
Computer Engineering & Computer Science Department
Speed School of Engineering
University of Louisville
Louisville, KY 40292

EDUCATION:

Ph.D., Computer Science & Engineering
April 2015
University of Louisville, *Louisville, Kentucky*

M.S., Information System
June 2009
Wuhan University, *Wuhan, China*

B.S., Electrical Engineering
June 2007
Huazhong University of Science and Technology, *Wuhan, China*

PUBLICATIONS:

1. **S. Jiang** and H. Frigui and A. Calhoun, *"Text-independent Speaker Identification Using Fisher Vector Representation"*, to be submit.

2. **S. Jiang** and H. Frigui and A. Calhoun, *"Text-independent Speaker Identification Using Soft Bag-of-Words Feature Representation"*, International Journal of Fuzzy Logic and Intelligent Systems, 14(4), pp.240-248, 2015.

3. **S. Jiang** and H. Frigui and A. Calhoun, *"Semantic Indexing of Video Simulations for Enhancing Medical Care During Crises"*, 11th International Conference on Machine Learning and Applications (ICMLA), pp.520-525, 2012.

HONORS AND AWARDS:

1. Grosscurth Fellowship by University of Louisville, 2009-2011.

2. Scholarship for Excellent Postgraduate, Wuhan University, 2008.

3. Wuhan University Graduate Fellowship, 2007 - 2008.

4. Scholarship for Excellent Undergraduate, HUST, 2004 - 2007.